2020

A Drift-Resilient and Degeneracy-Aware Loop Closure Detection Method for Localization and Mapping In Perceptually-Degraded Environments

Kamak Ebadi

Santa Clara University

Department of Electrical and Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Kamak Ebadi

ENTITLED

A Drift-Resilient and Degeneracy-Aware Loop Closure Detection Method for Localization and Mapping In Perceptually-Degraded Environments

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN ELECTRICAL AND COMPUTER ENGINEERING

Sally L. Wood, Ph.D.
Thesis Advisor

Christopher A. Kitts, Ph.D.

Aleksandar Zecevic, Ph.D.

Maryam Khanbaghi, Ph.D.

Ali-akbar Agha-mohammadi, Ph.D.

A Drift-Resilient and Degeneracy-Aware Loop Closure Detection Method for Localization and Mapping In Perceptually-Degraded Environments

By

Kamak Ebadi

Dissertation

Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical and Computer Engineering
in the School of Engineering at
Santa Clara University, 2020

Santa Clara, California

"The future is independent of the past given the present."

The Markov property.

"so that his place shall never be with those cold and timid souls who neither know victory or defeat."

Theodore Roosevelt

To my beloved Aram.

Acknowledgments

I would like to thank the people whose support made this dissertation possible. First and foremost, I would like to thank the members of my Ph.D. committee for their support throughout my degree. In particular, I would like to extend my gratitude to my advisor, Professor Sally Wood, for her mentorship and all the insightful and intellectual discussions over the past several years. I have benefited tremendously from her wisdom and valuable guidance.

I am thankful to Professor Christopher Kitts, the director of Robotics Systems Laboratory, for introducing me to the world of robotics and space systems, and for his continued support throughout my degree. I enjoyed our many robotic field experiments in lake Tahoe and NASA Ames research center that sparked my interest in autonomous navigation of mobile robots.

I am grateful to many people at NASA Jet Propulsion Laboratory, California Institute of Technology, where I carried out a substantial part of my Ph.D. research. My heartfelt gratitude to my greatest life mentor, Dr. Firouz Naderi, the former Mars program manager and director for solar system exploration at NASA JPL. He has been an immense source of support and inspiration to me, and I have nothing but the utmost appreciation and admiration for all the thought-provoking discussions and invaluable professional and life lessons that he has taught me over the years.

I would like to extend my gratitude to Dr. Curtis Padgett, the group supervisor of Maritime and Aerial Perception group, for the remarkable opportunity to join his group at NASA JPL to collaborate on some cutting-edge research projects which also supported my Ph.D. research. Our weekly technical meetings and discussions, and his guidance and suggestions have greatly contributed to my professional development.

Special thanks to Dr. Ali Agha, the Aerial Robotic Mobility group lead, for all the valuable advice and guidance on the several research projects that I collaborated with him. I am a better researcher because of him. Particularly, I appreciate the opportunity to collaborate with him on the DARPA Subterranean Challenge project. This has been an amazing endeavour to be part of a team of world-class researchers and engineers on such a groundbreaking robotic challenge that is aimed at pushing the boundaries of autonomy, robotics, and AI.

I am grateful to Dr. Amir Rahmani, the group supervisor of Maritime and Multi-Agent Autonomy group, for his guidance, mentorship and the "rounds of coffee and complements" throughout the past several years. No matter how busy he was, he always found time to listen to my concerns or questions, and to provide invaluable guidance and advice on both professional and personal levels.

I am thankful to Dr. Saptarshi Bandyopadhyay, for teaching me to always look on the bright side of life, and for his encouraging words, particularly, the word "sitzfleisch".

I thank Dr. Benjamin Morrell for all the great brainstorming sessions, and his continued care and support during my collaboration with him and his team. I have learned tremendously from his technical and leadership skills.

Special thanks to Professor Luca Carlone, the director of the MIT SPARK lab, for all the valuable discussions and his continued support of the team which lead to development of novel autonomy solutions and several successful publications. I appreciate all

the valuable lessons that I have learned from him throughout my collaboration on the DARPA Subterranean Challenge.

My biggest thanks is to my parents, my mom, Leila Saffarian, and my dad, Mohammad Ebadi, for always supporting my endeavors and giving me the confidence to pursue my dreams. My dad sparked my curiosity about space exploration when I was a child, and he is the reason that I decided to pursue a career in space robotics. He will live forever in my heart. I am grateful to my closest and oldest friend, my brother Bobby Ebadi, for believing in me before I believed in myself, and for helping me to stay on track toward my goals. His continued care and support is the reason for who I am today.

I am extremely grateful to my uncle and his wife, Reza and Cheryl Saffarian, for being an immense source of support, and inspiration in my life. I appreciate them for welcoming me when I first arrived in the U.S., and for being there for me every step of the way.

Above all, I owe great thanks to my wife, Aram Hamidi, for her love and never-ending support. I am deeply indebted to her for her sacrifices, for her patience and understanding, for being my rock, for all the late nights, for keeping me sane, and for believing in me and my dreams. I owe you everything.

A Drift-Resilient and Degeneracy-Aware Loop Closure Detection Method for Localization and Mapping In Perceptually-Degraded Environments

Kamak Ebadi

Department of Electrical and Computer Engineering
Santa Clara University
Santa Clara, California
2020

Abstract

Enabling fully autonomous robots capable of navigating and exploring unknown and complex environments has been at the core of robotics research for several decades. Mobile robots rely on a model of the environment for functions like manipulation, collision avoidance and path planning. In GPS-denied and unknown environments where a prior map of the environment is not available, robots need to rely on the onboard sensing to obtain locally accurate maps to operate in their local environment. A global map of an unknown environment can be constructed from fusion of local maps of temporally or spatially distributed mobile robots in the environment.

Loop closure detection, the ability to assert that a robot has returned to a previously visited location, is crucial for consistent mapping as it reduces the drift caused by error accumulation in the estimated robot trajectory. Moreover, in multi-robot systems, loop closure detection enables finding the correspondences between the local maps obtained by individual robots and merging them into a consistent global map of the environment. In ambiguous and perceptually-degraded environments, robust detection of intra- and inter-robot loop closures is especially challenging. This is due to poor illumination or lack-thereof, self-similarity, and sparsity of distinctive perceptual landmarks and fea-

tures sufficient for establishing global position. Overcoming these challenges enables a wide range of terrestrial and planetary applications, ranging from search and rescue, and disaster relief in hostile environments, to robotic exploration of lunar and Martian surfaces, caves and lava tubes that are of particular interest as they can provide potential habitats for future manned space missions.

In this dissertation, methods and metrics are developed for resolving location ambiguities to significantly improve loop closures in perceptually-degraded environments with sparse or undifferentiated features. The first contribution of this dissertation is development of a degeneracy-aware SLAM front-end capable of determining the level of geometric degeneracy in an unknown environment based on computing the Hessian associated with the computed optimal transformation from lidar scan matching. Using this crucial capability, featureless areas that could lead to data association ambiguity and spurious loop closures are determined and excluded from the search for loop closures. This significantly improves the quality and accuracy of localization and mapping, because the search space for loop closures can be expanded as needed to account for drift while decreasing rather than increasing the probability of false loop closure detections.

The second contribution of this dissertation is development of a drift-resilient loop closure detection method that relies on the 2D semantic and 3D geometric features extracted from lidar point cloud data to enable detection of loop closures with increased robustness and accuracy as compared to traditional geometric methods. The proposed method achieves higher performance by exploiting the spatial configuration of the local scenes embedded in 2D occupancy grid maps commonly used in robot navigation, to search for putative loop closures in a pre-matching step before using a geometric verification. The third contribution of this dissertation is an extensive evaluation and analysis of performance and comparison with the state-of-the-art methods in simulation and in real-world, including six challenging underground mines across the United States.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Mobile robots rely on a model of the environment for manipulation, collision avoidance and path planning. In GPS-denied and unknown environments where a prior map of the environment is not available, robots need to rely on the onboard sensing to obtain locally accurate maps to perform their tasks. However, functions such as efficient long term path planning require a metrically and topologically accurate global map of the environment, where the map can be constructed from fusion of local maps obtained by temporally or spatially distributed mobile robots exploring the unknown the environment.

A robot navigating an unknown environment needs to construct a map of the environment while using the same map for localization. This is referred to as simultaneous localization and mapping (SLAM), that is a challenging problem as both the map and robot pose are unknown. SLAM can be cast as an optimization process where the objective is to infer the most probable map of the environment given the set of all measurements, and to simultaneously infer the robot pose within the created map. The

accumulation of errors in the estimated robot poses can lead to an unbounded drift in the estimated robot trajectory for an assumed unlimited operation time. Loop closure detection, the problem of correctly ascertaining that a robot has returned to a previously visited location, is a crucial component of any SLAM system that ensures previously visited locations are not remapped in incorrect global locations and reduces the accumulation of drift in robot trajectory between known landmarks.

Many SLAM systems rely on robot pose estimates within the constructed map to determine if a robot is revisiting a previously visited location [1–4]. The loop closure detection problem rapidly becomes more challenging in large-scale and ambiguous environments where sensor noise and non-linearities make the environment observations less reliable. As presented in Fig. 1.1, in these environments sensors must operate in off-nominal conditions; poor illumination or lack-thereof, dust and non-Lambertian surfaces render visual-SLAM approaches unreliable [5]; uneven and slippery terrains make wheel odometry inaccurate, while long and featureless corridors make lidar-based odometry and mapping prone to drift; finally, perceptual aliasing, the presence of many similar-looking corridors and intersections, induces spurious loop closures that can degrade the mapping results. Lidar-based SLAM has been a popular solution to mapping perceptually-degraded environments, as lidar sensors are active sensors that provide range data over a 360° horizontal field of view at a high temporal and spatial sampling rate and do not rely on any external light sources.

The traditional lidar-based loop closure detection methods rely on lidar scan registration [4,6–9] to identify potential loop closures as the robot traverses an unknown environment. One significant draw back of such methods is that loop closures candidates are identified based on their proximity to the estimated robot poses. The main underlying problem in these methods is that the likelihood used to identify loop closures depends on the estimated robot poses, thus, loop closures cannot be reliably detected if there are

large errors between true and estimated robot poses. Constraining the search for loop closures to only the neighbourhood of robot's estimated poses is not robust in the face of incremental errors in robot's ego-motion estimation, especially in perceptually-degraded environments with sparse salient perceptual features that could lead to noisy odometric estimates. For instance, a small heading error over long linear traverses could lead to substantial position errors in large-scale or long-term operations.

This dissertation is focused on improving localization and mapping in extreme and ambiguous environments (i.e., mines, caves, and lava tubes) that barely provide distinctive landmarks and geometrical features sufficient to establish global localization. In this context, new methods and metrics are developed to improve localization and mapping in perceptually-degraded environments by improving the reliability and robustness of lidar-based loop closures in single and multi-robot SLAM systems.

1.2 Research Contributions

1.2.1 A degeneracy-aware lidar-based SLAM front-end

This dissertation provides a formal definition of geometric degeneracy, and develops methods and metrics to enable a real-time degeneracy-aware SLAM front-end capable

Fig. 1.1: Some examples of challenges introduced by perceptually-degraded subterranean environments to robot perception systems.

of online determination of environment degeneracy. The proposed method is based on computing the Hessian associated with the optimal computed local transformation obtained from lidar scan registration. Using this crucial capability, ambiguous areas in an unknown environment that could lead to poor or spurious loop closures are determined and excluded from the search for loop closures. This significantly improves the quality and accuracy of detected intra- and inter-robot loop closures because the loop closure search spaces can be expanded as needed to account for drift in the estimated robot trajectory while decreasing rather than increasing the probability of false loop closure detections.

1.2.2 A drift-resilient semantic-geometric loop closure detection method

A drift-resilient loop closure detection method is developed based the 2D semantic and 3D geometric features extracted from lidar point cloud data to enable detection of loop closures with increased robustness and accuracy as compared to traditional methods. This multi-stage process does not rely on any additional sensors, and achieves higher performance and accuracy by exploiting a pre-matching step based on the semantic information embedded in 2D occupancy grid maps that are commonly used in robot navigation and motion planning. By constructing 2D occupancy grid maps from lidar point cloud data, more information about the shape and spatial configuration of the local environment is obtained from lidar scans. This information is used to improve place recognition, especially in perceptually-degraded environments where lidar-scan matching alone could lead to inaccurate or spurious loop closures due to sparsity of features and high level of similarity between individual lidar scans. Moreover, the proposed method is pose-invariant and it does not rely on estimated robot trajectory to identify loop closure candidates and thus, it is not affected by the accumulation of errors in the estimated robot trajectory.

Fig. 1.2: Examples of the perceptually-degraded underground environments used for validation and testing of the methods and metrics developed in this thesis. From left to right, Eagle Mine in Julian, CA, Arch Pocahontas Mine in Beckley, WV and Edgar Mine in Idaho Springs, CO.

1.2.3 Extensive real-world experiments

The third contribution of this dissertation is implementation of the proposed methods on a multi-robot SLAM system and evaluation and analysis of performance and comparison with the state-of-the-art methods in a variety of indoor and challenging underground environments, including six challenging underground mines across the United States. Fig. 1.2 presents some examples of explored underground mines. Table 5.5 provides the list of explored underground mines across the United States.

1.3 Experimental Setup and Ground Truth

The robots used in the experiments are Husky-A200 series, equipped with an Intel NUC 7i7DNBE (4 $\times$ 1.9 GHz, 32 GB RAM) processor, an Intel RealSense D435 RGB-

Table 1.1: List of the explored underground mines.

Name of the mine	Autonomously Traversed Distance	Type of mine	Location
Arch Pocahontas Mine	1100 m	Coal Mine	Beckley, WV
Beckley Exhibition Mine	1000 m	Coal Mine	Beckley, WV
Bruceton Safety Research Mine	1400 m	Coal Mine	Pittsburgh, PA
Bruceton Experimental Mine	700 m	Coal Mine	Pittsburgh, PA
Highland Mine	1400 m	Coal Mine	Logan, WV
Eagle Mine	500 m	Gold Mine	Julian, CA

D camera, and a VLP-16 Puck Lite lidar [10]. Using the VLP-16 lidar scanner, up to $30,000$ points/scan are obtained across a 360° horizontal and a 30° vertical field of view. With 16 scanning channels, VLP-16 provides a vertical angular resolution of 2.0° and a horizontal resolution of 0.1° to 0.4°.

The RGB-D camera operates at 60 FPS and provides images at 1280×720 resolution. The camera is used to detect objects of interest and to estimate their relative position with respect to the robot. The FOV of the RGB camera is $69.4° \times 42.5° \times 7°$, and the depth camera has a FOV of $91.2° \times 65.5° \times 100.6°$ and an effective depth range of 0.2m to 10m.

In multi-robot experiments, a centralized architecture is developed where a base station is used for receiving the local maps and trajectories created by each robot, and to align and merge them into a global map of the explored environment. The base station is an Intel Hades Canyon NUC8i7HVKVA (4×1.9 GHz, 32 GB RAM). Obtaining ground truth trajectory and maps in large-scale underground environments is not always possible. As presented in Fig. 1.2, two fiducial markers with known positions in the world coordinate system are used for robot pose calibration at the entrance to the mines. Relying on the onboard RGB-D camera, each robot uses the AprilTags library [11] to find and localize the fiducial markers in order to calibrate its initial pose in a world coordinate system.

Furthermore, an object-based approach is developed to obtain the ground truth trajectory in each test environment. The surveyed locations of known objects placed at different locations in the environment are used to obtain a proxy for the ground truth trajectories of the robots. This is achieved by enforcing the ground truth locations of objects in the best trajectory of each robot, and use the resulting optimized trajectory as ground truth.

1.4 Dissertation Outline

Chapter 2, presents the necessary background information and related work on single and multi-robot SLAM systems with focus on perceptually-degraded and subterranean environments. This includes the algorithms currently used and their limitations, and gaps in the state of the art to highlight the contribution of this dissertation.

Chapter 3, presents development of a lidar-based SLAM front-end that relies on a two-stage scan-to-scan and scan-to-submap matching process to achieve more accurate odometric estimates. Through extensive experiments in real-world subterranean environments performance and accuracy of the proposed lidar-based front-end is qualitatively and quantitatively analyzed.

In chapter 4, metrics and methods are developed to enable real-time determination of geometric degeneracy in unknown environments based on computing the Jacobian associated with the optimal computed local transformations obtained from lidar scan registration. Through extensive simulations and real-world experiments the reliability and performance of the developed methods are analyzed in a variety of challenging environments.

Chapter 5 addresses the problem of loop closure detection in lidar-based SLAM systems and presents development and analysis of a multi-stage loop closure detection pipeline that relies on 2D semantic and 3D geometric features extracted from lidar point cloud data. Through real-world experiments in a variety of perceptually-degraded underground and indoor environments the performance, robustness and reliability of the proposed method is evaluated, and localization and mapping results are compared with commonly used state-of-the-art methods. Chapter 6 concludes the dissertation and discusses future research directions.

Chapter 2

Background and Related Work

2.1 Mobile Robot Navigation

The ability to autonomously navigate an unknown environment is a crucial capability for any mobile robot. The fundamental objective in autonomous navigation is to arrive at a goal location while avoiding the hazards and collision with obstacles in the scene. In order to safely navigate the environment, an autonomous robot is required to answer three key questions.

- Where am I: to establish robot pose in a global reference frame.
- Where is the target: to establish target's location in a global reference frame.
- How do I get there: to plan the safest path to the target.

If the environment is known, and a prior map exists, the robot can establish its global pose by registering measurements obtained from onboard perception system to the map. In an unknown environment, an accurate model of the local scene must be constructed by relying on the onboard perception system. This introduces many uncertainties as the

true state of the robot is unknown, and can only be estimated by relying on noisy sensor measurements. Moreover, the motion model of the robot can only be approximated as actuators may not accurately carry out each command, either due to inaccuracy of the actuators or the specific characteristics of the environment (e.g., wheel slippage on a slippery terrain).

2.2 Sensor Modalities in SLAM

Over the past two decades, many SLAM solutions have been proposed to recover the most probable representation of the map and the robot pose based on different sensor modalities, including vision [3], visual-inertial [12–14], and thermal-inertial [15]. Vision-based localization and place recognition is a well studied problem in the robotics and computer vision literature [16], and while great progress has been made in visual loop closure detection [3, 17], in perceptually-degraded environments vision-based methods are faced with many challenges including reduced visibility (i.e., inconsistent illumination or lack-thereof, fog, and dust), and varying viewing angles that can lead to significant variations in the appearance of a scene between different visits [18, 19]. Moreover, sparsity of salient perceptual features can lead to data association ambiguity and perceptual aliasing. To overcome these challenges, cameras and lidars are often used in conjunction due to their complementary nature [20, 21].

Lidar-based SLAM has been a popular solution to map perceptually-degraded environments, as lidar sensors provide range data over a 360° horizontal field of view at a high temporal and spatial sampling rate, and do not rely on external light sources. Nuchter et al. [22] present a lidar-based 6D SLAM algorithm that is used on a mine inspection robot. In order to achieve real-time performance, the authors propose a fast filtering method based on combining a median and a reduction filter to achieve significant data

reduction in lidar scans while maintaining the surface structure in subterranean environments. Zhang et al [12,23] present lidar odometry and mapping (LOAM), that relies on feature point extraction and matching to achieve real-time performance while minimizing motion distortions. While LOAM achieves good performance, it currently does not recognize loop closures. Open loop systems can accumulate drift due to matching and motion estimation errors.

Accurate loop closure detection, the ability to correctly assert that a robot has returned to a previously visited location, is a fundamental component of simultaneous localization and mapping systems to limit and reduce the drift in the estimated robot trajectory between two known locations. The earliest loop closure detection systems were based on registration of the shape of the laser scans [24–26]. Cox et al. [24] proposed a method designed for use in structured office or factory environments where points were aligned with line segments extracted from a known environment representation. Gutmann et al. [25] developed a scan-to-scan alignment approach that relied on the extraction of line segments from reference scans. Lu et al. [26] developed two iterative scan matching algorithms based on the Iterative Closest Points (ICP) [27] algorithm which do not require feature extraction or segmentation, thus, improving the point-to-line matching method of Cox et al. with a point-to-point matching approach. Moreover, similar to feature-based registration of 2D images [28], several methods based on extracting features from lidar point cloud [12,29–31] were proposed where the point cloud alignment is achieved through matching the computed feature descriptors.

Over the past decades the ICP algorithm has been commonly used for lidar scan registration and loop closure detection [4,6–9]. In order to detect a loop closure, a lidar scan is registered to previously obtained lidar scans to find a match. In large-scale or long-term operations where a large number of lidar scans are obtained this method can become increasingly computationally expensive, and also increases the probability

of spurious or inaccurate loop closures in ambiguous and featureless environments. In order to improve the performance and accuracy of loop closures, a common approach is to constrain the search for loop closures to a fixed radius centered at estimated robot poses. While this method is effective in reducing the computational load and spurious loop closures, it could lead to missed loop closure opportunities when the accumulation of errors in lidar odometry leads to significant drift in the robot trajectory.

Ji et al. [32] present a 3D lidar mapping algorithm similar to LOAM that relies on segment based mapping and place recognition [33, 34] in 3D point clouds to detect loop closures in structured street environments. Shan et al. [6] propose an extension to LOAM and present the lightweight and ground-optimized (LeGO-LOAM) algorithm that relies on the ground segmentation capability to discard points that may represent unreliable features. While their proposed method maintains the local consistency of the ground plane between consecutive frames, it does not use a global ground constraint that makes it susceptible to the accumulation of rotational error. In large-scale underground environments with uneven terrain, this could lead to large errors in the estimated robot poses and trajectories. In LeGO-LOAM the search for loop closures is performed locally based on the estimated robot poses. This could lead to missed loop closure opportunities when drift in the robot trajectory is significant. Hess et al. [35], present Cartographer with real-time 2D mapping and loop closure capability. By combining a few consecutive lidar scans into a local submap, loop closures are detected by matching a large set of submaps. Upon detection of loop closure candidates, outlier loop closures are rejected by relying on the Huber loss in Sparse Pose Adjustment [36].

2.3 SLAM Overview

Enabling fully autonomous mobile robots, capable of navigating unknown environments has been at the core of robotics research for several decades. A fundamental requirement for realizing autonomous navigation is simultaneous localization and mapping. SLAM is a major, yet relatively new field in robotics [37, 38], that can be formulated as a process by which a mobile robot can build a map of an unknown environment and at the same time use this map to deduce it's location. Both the robot trajectory and landmark locations are unknown and need to be estimated in real-time. This can be formulated as computing the probability distribution over all possible robot states and the map at time k, from the set of sensor measurements $z_{0:k}$, and control inputs $u_{0:k}$ as given by:

$$P(x_k, m|z_{0:k}, u_{0:k}, x_0), \tag{2.1}$$

where x_k is the state vector describing the robot pose, m is the map, $u_{0:k}$ is the history of control inputs, and x_0 is the known initial robot pose.

The first SLAM methods [39, 40] relied on an Extended Kalman Filter (EKF) to estimate the robot pose and location of landmarks from sensor measurements, and the robot's motion model. The EKF SLAM is based on the recursive Bayesian state estimation theory where the robot state is estimated through an iterative three-step process of prediction, observation, and update. A major drawback of EKF SLAM is that, in large-scale or long-duration operations the algorithm will become increasingly more expensive in terms of memory and computational requirements as the number of mapped features increases [41].

Subsequently, SLAM algorithms based on the Extended Information Filter (EIF) [41], and the Sparse Extended Information Filter [42] were proposed that enabled higher effi-

ciency than EKF-based methods. However, these methods did not allow relinearization of observations which made them susceptible to high noise in sensor measurements. To address the nonlinearities and the sensor noise, Grisetti et al. [43] propose a SLAM solution based on particle filters where the basic idea is to use a set of discrete weighted particles to simulate the posterior probability of the estimated robot state. While particle filter-based SLAM solutions enable higher efficiency as compared to EKF-based methods, they are susceptible to the particle-depletion problem [44] that is, a lack of particles corresponding to the true state of the robot. To address the shortcomings of EKF and particle filter-based methods, the first graph-based SLAM solutions emerged between 1997 and 2007 [45–47], which enabled relinearization of the obtained measurements. Using a pose graph representation, the pose of the robot can be estimated by solving the graph and finding the best graph configuration that is most consistent with the obtained measurements. Given that in pose graph architecture the graph contains the history of all previous measurements and information, the method provide a much higher accuracy as compared to traditional SLAM methods discussed in Section 2.3.

Graph-based SLAM has been the method of choice in single and multi robot systems [45, 48–53]. A graph representation not only reduces the communication bandwidth required to communicate the local maps between agents or to a base station, but also enables a search for loop closures based on the similarity of instances. In the context of vision-based mapping, Erinc et al. [50] propose a method to merge topological graphs by analyzing the algebraic connectivity as a metric to determine the gain obtained by merging multiple maps together. Huang et al. [54] propose an algorithm for merging embedded topological maps. Relying on the concise description of the navigability of an environment embedded in topological maps, the vertices of the maps are embedded in a metric space. The authors propose an algorithm that relies on both the structure and the geometry of topological maps to determine the best correspondence between

maps with single or multiple overlapping regions. However, for the map alignment to be successful their method requires a high topological similarity between sub-maps. Saeedi et al. [55] capture the local structure of the environment from metric information and compute probabilistic topological graphs by extracting the Voronoi diagram of each input. The diagrams are then merged by maximizing a cross correlation value through edge matching.

Choudhary et al. [48, 49] present a distributed multi-robot SLAM solution that relies on object landmarks for localization and mapping and shows that object-based representations reduce the memory requirements and information exchange among robots, compared to point-cloud-based representations. Bonanni et al. [56] present a solution for merging three-dimensional maps represented as pose graphs of point clouds, where a robot is localized in a reference map by using the data from another map as observations. Lazaro et al. [53] propose a graph-based SLAM solution that utilizes condensed measurements to exchange map information between multiple robots. In order to align the maps, each laser scan in one map is registered to all laser scans in the other map and best matches are entered into a pool of candidates where a RANSAC algorithm is used to reject outlier matches. A bottleneck of this approach is that loop closure detections become computationally more expensive in long-term and large-scale operations.

2.3.1 SLAM front-end

The primary function of a graph-based SLAM front-end is to produce relative motion estimates between consecutive robot poses (odometry) and non-consecutive poses (loop closures), to create a pose graph by relying on the measurements obtained by robot's onboard perception system. As illustrated in Fig. 2.1, in order to represent a robot trajectory and the map of the environment, a graph-based formulation is used [57] where

a robot trajectory is represented as a pose graph, where every node in the graph corresponds to a robot pose, and every edge connecting two nodes in the graph expresses the relative pose measurement between corresponding robot poses. We denote the trajectory of robot α by the sequence of N_α random variables $\mathbf{x}_\alpha \doteq [\mathbf{x}_{\alpha_i}]_{i=0:N_\alpha}$, where $\mathbf{x}_{\alpha_i} \doteq [\mathbf{R}_{\alpha_i}, \mathbf{t}_{\alpha_i}] \in \mathrm{SE}(3)$ is the robot pose associated to the i-th node in the graph and is represented as a 3D transformation that includes a rotation $\mathbf{R}_{\alpha_i} \in \mathrm{SO}(3)$, and a position $\mathbf{t}_{\alpha_i} \in \mathbb{R}^3$.

The set of all raw lidar scans obtained along the trajectory of robot α is denoted by $z_\alpha = \{z_{\alpha_i}\}_{i=0:N_\alpha}$, where $z_{\alpha_i} = \{p_k\}_{k=1:N_z}$ is the i-th lidar scan represented as a set of spatial measurements $p_k \in \mathbb{R}^3$ in the local lidar coordinate system $\mathcal{L}_i$. The measurement model is denoted by

$$z_{\alpha_i} = h(\mathbf{x}_{\alpha_i}, m_\alpha) + w_{\alpha_i}, \tag{2.2}$$

where $h(.)$ returns the vector connecting the current lidar pose to the environment along each lidar ray, m_α represents part of the actual local environment explored by robot α, and $w_{\alpha_i} \sim \mathcal{N}(0, \Sigma_\alpha)$ is assumed to be a Gaussian noise with zero-mean and covariance Σ_α.

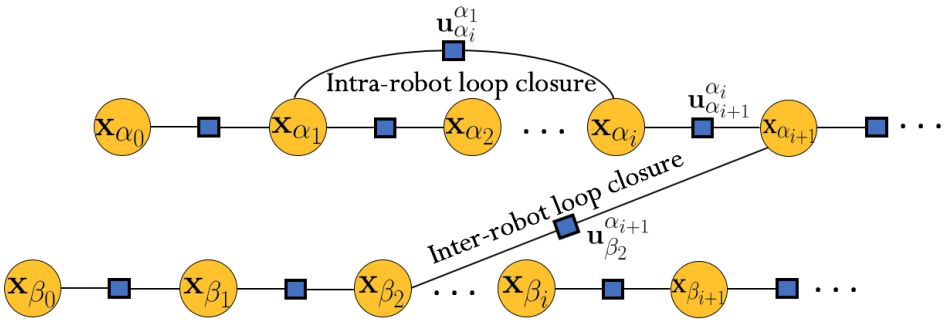

Fig. 2.1: Illustration of pose graph SLAM. The yellow nodes correspond to the unknown robot poses, while blue squares represent the relative 3D pose measurements obtained from lidar scan matching.

The lidar-based odometric measurements are obtained by computing the relative pose transformation between two consecutive lidar scans. Most lidar odometry methods rely on the ICP algorithm or its variants [58], or feature-based approaches [29] to recover the pose of the robot with respect to its initial pose. The ICP-based lidar scan registration is by far the most popular scan registration method, which aims to minimize a predefined error function in an iterative optimization process, through which new correspondences are found at each iteration of the process and a rigid body transformation is computed to minimize the error function.

An ICP-based lidar odometry system finds the 3D transformation that minimizes the mean squared error between the set of corresponding points by using a least squares optimization process. In this dissertation the Generalized Iterative Closest Point (GICP) algorithm developed by Segal et al. [59] is used. GICP is a unifying framework of the previously proposed ICP methods that combines the point-to-point and plane-to-plane algorithms into a single probabilistic framework to have a better accuracy over the original ICP algorithm. Given two corresponding point sets $P = \{p_i\}_{i=1:N_k}$, $Q = \{q_i\}_{i=1:N_k}$, where N_k denotes the number of points, the ICP algorithm seeks to find the translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $\mathbf{R} \in SO(3)$ that minimizes the sum of the squared distances between the corresponding points as given by

$$\hat{\mathbf{R}}, \hat{\mathbf{t}} = \underset{\mathbf{R} \in SO(3), \mathbf{t} \in \mathcal{R}^3}{\operatorname{argmin}} \sum_{k=1}^{N_k} ||\mathbf{R}p_k + \mathbf{t} - q_k||^2. \tag{2.3}$$

The minimization problem can be simplified by finding the center of mass in each point cloud and subtracting the corresponding center of mass from every point in the two point sets. The center of mass of each point cloud is given by

$$\bar{p} = \frac{1}{|P|} \sum_i p_i, \tag{2.4}$$

similarly for Q,

$$\bar{q} = \frac{1}{|Q|} \sum_{j} q_j. \tag{2.5}$$

By subtracting the corresponding center of mass from every point in the two point sets the normalized points are obtained:

$$p_{k'} = p_k - \bar{p} \Rightarrow p_k = p_{k'} + \bar{p}, \tag{2.6}$$

and,

$$q_{k'} = q_k - \bar{q} \Rightarrow q_k = q_{k'} + \bar{q}. \tag{2.7}$$

Hence (2.3) can be written as

$$\sum_{k=1}^{N_k} ||\mathbf{R}p_k + \mathbf{t} - q_k||^2 = \sum_{k=1}^{N_k} ||\mathbf{R}(p_{k'} + \bar{p}) + \mathbf{t} - (q_{k'} + \bar{q})||^2 \tag{2.8}$$

$$= \sum_{k=1}^{N_k} ||\mathbf{R}p_{k'} - q_{k'} + (\mathbf{R}\bar{p} - \bar{q} + \mathbf{t})||^2. \tag{2.9}$$

now if,

$$\mathbf{t} = \bar{q} - \mathbf{R}\bar{p}, \tag{2.10}$$

we can simplify (2.10) as given by

$$\sum_{k=1}^{N_k} ||\mathbf{R}p_k + \mathbf{t} - q_k||^2 = \sum_{k=1}^{N_k} ||\mathbf{R}p_{k'} - q_{k'}||^2. \tag{2.11}$$

Expanding (2.11):

$$\sum_{k=1}^{N_k} ||\mathbf{R}p_{k'} - q_{k'}||^2 = RR^T \sum_{k=1}^{N_k} ||p_{k'}||^2 - 2tr(R\sum_{k=1}^{N_k} p_{k'}q_{k'}^T) + \sum_{k=1}^{N_k} ||q_{k'}||^2. \tag{2.12}$$

where $tr(.)$ denotes matrix trace operation. Given $RR^T = I$,

$$\sum_{k=1}^{N_k} ||\mathbf{R}p_{k'} - q_{k'}||^2 = \sum_{k=1}^{N_k} ||p_{k'}||^2 - 2tr(R\sum_{k=1}^{N_k} p_{k'}q_{k'}^T) + \sum_{k=1}^{N_k} ||q_{k'}||^2. \tag{2.13}$$

Let the sum of products of the normalized points be denoted by

$$M = \sum_{k=1}^{N_k} p_{k'}q_{k'}^T. \tag{2.14}$$

Hence, the problem of minimization of (2.13) reduces to that of maximization of $tr(RM)$. Taking the singular value decomposition (SVD) of matrix M,

$$M = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T \tag{2.15}$$

where U and V are orthogonal matrices and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are singular values of M. It can be proved [60] that if $Rank(M) = 3$, the optimal solution to minimize equation (2.3), is unique and is given by

$$\hat{\mathbf{R}} = VU^T \tag{2.16}$$

$$\hat{\mathbf{t}} = \bar{p} - \hat{\mathbf{R}}\bar{q} \tag{2.17}$$

In real world scenarios the correct correspondences are unknown, thus, determining the optimal relative rotation and translation in one step is not possible. Given a limited

number of iterations, the process of determining the set of corresponding points and the 3D transformation that best aligns them is iteratively repeated until the algorithm reaches a termination criteria. Four termination criteria can be considered for the transformation refinement process [61]:

1. **Maximum number of iterations** indicates the maximum number of iterations for the ICP algorithm to search for the corresponding points and refine the estimated 3D transformation between the them. While increasing the number of ICP iterations could improve the convergence of the algorithm, this comes at a higher computational cost which could affect the ability of the front-end system to run in real-time. Considering lidar scans are obtained at a high temporal frequency (e.g., 10 Hz to 20 Hz), the SLAM front-end needs to run at the same frequency, or otherwise some lidar scans will be dropped which could lead to poor estimation of robot pose and motion.

2. **Absolute transformation threshold** indicates that the ICP algorithm must be stopped when the currently estimated 3D transformation is far away from the initial transformation. This criteria helps to reduce the drift and divergence in the odometric estimates by using the intuition that two consecutive lidar scans are expected to be within a certain range of displacements from each other, thus, transformation that are outside the range need to be rejected.

3. **Relative transformation threshold** specifies the minimum transformation difference from one iteration to next, that is considered small enough for the optimizer to have converged.

4. **Maximum number of similar iterations** helps remedy a corner case, where the relative transformation threshold criteria might temporarily seem to have converged, while it is actually oscillating in a local minima and has a chance of escaping

from it and converging into the global minima or a better local minima. For this reason, the optimizer is allowed to spend a certain number of iterations around a minimum point before considering it converged.

Let $\mathbf{u}_{\alpha_{i+1}}^{\alpha_i} \doteq [\mathbf{R}_{\alpha_{i+1}}^{\alpha_i}, \mathbf{t}_{\alpha_{i+1}}^{\alpha_i}]$, denote an odometric estimate obtained by the ICP algorithm that encodes the relative 3D motion between two lidar scans z_{α_i} and $z_{\alpha_{i+1}}$, where $\mathbf{t}_{\alpha_{i+1}}^{\alpha_i} = [t_x, t_y, t_z]^T$ is the 3D translation vector, and the 3D rotation $\mathbf{R}_{\alpha_{i+1}}^{\alpha_i}(\theta_x, \theta_y, \theta_z)$ can be defined in terms of Euler angles θ_x, θ_y, and θ_z that denote roll, pitch and yaw respectively.

A yaw is a rotation of θ_z about the z-axis as given by

$$\mathbf{R}_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.18}$$

A pitch is a rotation of θ_y about the y-axis as given by

$$\mathbf{R}_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}. \tag{2.19}$$

A roll is a rotation of θ_x about the x-axis as given by

$$\mathbf{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}. \tag{2.20}$$

A 3D rotation matrix can be formed by multiplying the yaw, pitch, and roll rotation

matrices as given by

$$\mathbf{R}(\theta_x, \theta_y, \theta_z) = \mathbf{R}_z(\theta_z)\,\mathbf{R}_y(\theta_y)\,\mathbf{R}_x(\theta_x) \tag{2.21}$$

The relative pose measurement between two consecutive lidar scans is presented as a homogeneous transformation matrix as given by

$$\mathbf{u}_{\alpha_{i+1}}^{\alpha_i} = \left[\begin{array}{c:c} \mathbf{R}_{\alpha_{i+1}}^{\alpha_i}(\theta_x, \theta_y, \theta_z) & \mathbf{t}_{\alpha_{i+1}}^{\alpha_i} \\ \hdashline 0 \quad 0 \quad 0 & 1 \end{array}\right]$$

2.3.2 Drift and loop closure detection

Over the past decades, loop closure detection has become an essential part of SLAM systems. SLAM systems that rely on the onboard perception system to only provide odometry can inevitably accumulate significant drift in real world applications. In the absence of any prior map of the environment that can be used for global localization, the accumulation of errors from lidar scan matching can lead to an unbounded drift in the estimated robot trajectory for an assumed unlimited time. This drift is inherent to any odometry system and can only be limited and subsequently reduced by detection of loop closures when a robot returns to a previously visited location. Accurate loop closure detection, the ability to correctly assert that a robot has returned to a previously visited location, is a fundamental component of any SLAM system to limit and reduce the drift in the estimated robot trajectory between two known locations. In a single-robot SLAM system, each intra-robot loop closure represents a nonlinear constraint on the robot's local pose graph. In a multi-robot SLAM system, inter-robot loop closures are crucial to detect the correspondences between multiple robot trajectories. With well

constrained pose graphs the robot trajectories can be optimally adjusted by formulating the problem as the maximum a-posteriori (MAP) likelihood estimation of relative poses in the pose graph, given the set of all loop closure constraints.

In order to detect a loop closure in a lidar-based graph SLAM, a new lidar scan is registered to all other lidar scans associated with the historical nodes in the graph to find a match [4]. The ICP algorithm has been commonly used for lidar scan registration and loop closure detection in lidar-based SLAM systems [7–9]. In long-term or large-scale mapping in perceptually-degraded environments closing loops by matching a lidar scan to all previously obtained scans is computationally expensive, and could lead to increased numbers of spurious loop closures. Let α_i be the most recently created key-node in the local pose graph associated with the current pose $\mathbf{x}_{\alpha_i}$ of the robot α. A previous key-node $\alpha_{j=1:i-1}$ is considered a loop closure candidate if it satisfies three conditions. The first condition is if the robot pose $\mathbf{x}_{\alpha_j}$ associated with the key-node α_j is within a fixed distance D_r from $\mathbf{x}_{\alpha_i}$:

$$\left\|\mathbf{x}_{\alpha_i} - \mathbf{x}_{\alpha_j}\right\| < D_r, \tag{2.22}$$

where $\|.\|$ denotes the Euclidean distance, and D_r is the loop closure search radius. The Second condition is the distance in terms of the number of key-nodes in the graph between α_i and α_j:

$$\alpha_i - \alpha_j > D_k, \tag{2.23}$$

where D_k is the number of nodes in the local pose graph between α_i and α_j. Since drift in the estimated robot trajectory is minimal between consecutive key-nodes and larger between non-consecutive key-nodes, this condition helps reduce the computational load associated with the search for loop closures by removing the most recent key-nodes

prior to the current key-node from loop closure consideration. The third condition is the distance of α_i in terms of the number of key-nodes in the pose graph from the last successful loop closure key-node:

$$\alpha_i - \alpha_l > D_l, \tag{2.24}$$

where α_l is the most recent loop closure key-node in the pose graph, and D_l is the distance in terms of the number of key-nodes between α_i and α_l. This criteria reduces the computational load associated with the search for loop closures by ensuring that once a loop is successfully closed, the search for subsequent loop closures is suspended until a minimum number of new key-nodes are instantiated in the pose graph. The idea behind this criteria is that, once the drift in robot trajectory is minimized following a loop closure detection, subsequent loop closures can be delayed until drift is accumulated in robot trajectory again.

Let β_j be a historical key-node in the graph that satisfies the loop closure criteria in 2.22. Using the corresponding lidar scans z_{α_i} and z_{β_j}, the relative 3D motion $\mathbf{u}_{\beta_j}^{\alpha_i}$ between the two key-nodes can be computed using the ICP algorithm and the quality of the alignment can be determined by evaluating the mean squared error between the aligned point clouds. If the alignment error is less than a threshold, the constraint $\mathbf{u}_{\beta_j}^{\alpha_i}$ is added as an edge between two key-nodes. When $\alpha = \beta$, it represents an intra-robot loop closure found in a local pose graph, otherwise, it represents an inter-robot loop closure found between two local pose graphs of robots α and β on the base station.

2.3.3 SLAM back-end

Let $\mathbf{u}_{o_\alpha}$ denote the set of all relative pose measurements between consecutive (odometry) and non-consecutive (intra-robot loop closures) poses of robot α respectively. The goal

of pose graph optimization, is to estimate both the unknown robot trajectory and the map of the environment from the set of all intra-robot relative pose measurements. Let $G_\alpha = \langle \mathbf{x}_\alpha, \mathbf{u}_\alpha, \mathbf{z}_\alpha \rangle$ denote the local pose graph of robot α comprised of robot trajectory $\mathbf{x}_\alpha$, the set of all odometry and intra-robot loop closures $\mathbf{u}_\alpha \doteq [\mathbf{u}_{o_\alpha} \cup \mathbf{u}_{c_\alpha}]$, and the set of all key-scans $\mathbf{z}_\alpha$ obtained at each corresponding robot pose. We rely on Pose Graph Optimization (PGO) [57] to obtain the maximum a-posteriori (MAP) estimate of the robot trajectory as given by

$$\begin{aligned} \hat{\mathbf{x}}_\alpha &= \underset{\mathbf{x}_\alpha}{\operatorname{argmax}}\, p(\mathbf{x}_\alpha|\mathbf{u}_\alpha) \\ &= \underset{\mathbf{x}_\alpha}{\operatorname{argmax}}\, p(\mathbf{x}_\alpha) p(\mathbf{u}_\alpha|\mathbf{x}_\alpha), \end{aligned} \tag{2.25}$$

where $p(\mathbf{x}_\alpha)$ is the prior probability over $\mathbf{x}_\alpha$, and $p(\mathbf{u}_\alpha|\mathbf{x}_\alpha)$ is the measurement likelihood. When no prior knowledge is available over the robot trajectory, the term $p(\mathbf{x}_\alpha)$ is assumed to be a uniform distribution and can be dropped. This will reduce the MAP estimation problem to the maximum likelihood (ML) estimation as given by

$$\begin{aligned} \hat{\mathbf{x}}_\alpha &= \underset{\mathbf{x}_\alpha}{\operatorname{argmax}} \prod_{i=1}^{|\mathbf{u}_\alpha|} p(\mathbf{u}_{\alpha_i}|\mathbf{x}_{\alpha_i}) \\ &= \underset{\mathbf{x}_\alpha}{\operatorname{argmin}} -log\Big(\prod_{i=1}^{|\mathbf{u}_\alpha|} p(\mathbf{u}_{\alpha_i}|\mathbf{x}_{\alpha_i}) \Big) \\ &= \underset{\mathbf{x}_\alpha}{\operatorname{argmin}} -log\Big(\prod_{i=1}^{N_\alpha - 1} p(\mathbf{u}_{\alpha_{i+1}}^{\alpha_i}|\mathbf{x}_{\alpha_i}, \mathbf{x}_{\alpha_{i+1}}) \prod_{\mathbf{u}_{\alpha_j}^{\alpha_i} \in \mathbf{u}_{c_\alpha}} p(\mathbf{u}_{\alpha_j}^{\alpha_i}|\mathbf{x}_{\alpha_i}, \mathbf{x}_{\alpha_j}) \Big). \end{aligned} \tag{2.26}$$

Let $\mathbf{x}_{\alpha_j} = f(\mathbf{x}_{\alpha_i}, \mathbf{u}_{\alpha_j}^{\alpha_i}) + v_{i,j}$ be the robot motion model, where $f(.)$ implements the nonlinear motion between the unknown robot poses x_{α_i} and x_{α_j}, and $v_{i,j}$ is assumed to be a zero-mean Gaussian noise with information matrix $\Omega_{i,j}$. The ML trajectory estimate can be computed by minimizing the mismatch between the relative pose measurements

and the estimated robot poses as given by

$$\hat{\mathbf{x}}_\alpha = \underset{\mathbf{x}_\alpha}{\operatorname{argmin}} \Big(\sum_{i=1}^{N_\alpha - 1} \Big\| f(\mathbf{x}_{\alpha_i}, \mathbf{u}_{\alpha_{i+1}}^{\alpha_i}) - \mathbf{x}_{\alpha_{i+1}} \Big\|_{\Omega_{i,i+1}}^2 + \sum_{\mathbf{u}_{\alpha_j}^{\alpha_i} \in \mathbf{u}_{c_\alpha}} \Big\| f(\mathbf{x}_{\alpha_i}, \mathbf{u}_{\alpha_j}^{\alpha_i}) - \mathbf{x}_{\alpha_j} \Big\|_{\Omega_{i,j}}^2 \Big), \tag{2.27}$$

where $\|.\|_{\Omega_{i,j}}^2$ denotes the squared Mahalanobis distance.

To extend this to a multi-robot SLAM system, it is crucial to detect inter-robot loop closures to determine the overlapping regions between individual local pose graphs and to merge them into a consistent global pose graph. Assuming a team of two robots α and β exploring an unknown environment with known initial poses, after finding the set of all inter-robot loop closures $\mathbf{u}_I = \{\mathbf{u}_{\beta_j}^{\alpha_i}\}_{0 \leq i < N_\alpha, 0 \leq j < N_\beta}$, where i and j correspond to the indices of loop closure nodes, a global pose graph is constructed using the collection of all intra- and inter-robot relative pose measurements as denoted by $\mathbf{u}_g \doteq [\mathbf{u}_\alpha \cup \mathbf{u}_\beta \cup \mathbf{u}_I]$, and the collection of estimated robot trajectories as denoted by $\hat{\mathbf{x}}_g \doteq [\hat{\mathbf{x}}_\alpha, \hat{\mathbf{x}}_\beta]$. Similar to (2.25), the global pose graph optimization is formulated as a ML estimation problem

$$\overset{*}{\mathbf{x}}_g = \underset{\hat{\mathbf{x}_g}}{\operatorname{argmax}}\, p(\hat{\mathbf{x}}_g | \mathbf{u}_g), \tag{2.28}$$

where $\overset{*}{\mathbf{x}}_g$ denotes the best estimate of the global pose graph given the set of all intra- and inter-robot relative pose measurements. The global map of the environment is then obtained by projecting the point clouds associated with the nodes in the global pose graph into the common world coordinate system $\mathcal{W}$.

2.4 SLAM in Subterranean Environments

Over the past decades a number of efforts have focused specifically on localization and mapping in underground mines and tunnels. Thrun et al. [62] propose a SLAM algorithm for volumetric mapping of large underground mines. In order to reduce the drift in robot trajectories, the method uses a modified version of the ICP algorithm to estimate the correspondences between different lidar scans in order to detect loop closures. Tardioli et al. [63] propose a system for exploration of a tunnel using a robot team. The system relies on a feature-based robot LOcalization Module (LOM) that is responsible for global localization by relying on pre-existing two-dimensional laser-segmentation-based maps. Global localization is achieved by matching observations of the environment with the map features. In a similar work Tardioli et al. [64] propose an underground localization algorithm to enable autonomous navigation of a commercial dumper commonly used for underground construction. Localization is achieved by relying on semantic feature recognition in the tunnels that provide local information sufficient for a successful localization.

Zlot et al. [65] present a 3D SLAM solution consisting of a spinning 2D lidar and an industrial-grade MEMS IMU to map a 17km long underground mine. In order to reduce the drift in open-loop trajectory, the authors rely on a global optimization algorithm where the entire robot trajectory is optimized given the set of all constraints. Furthermore, a set of anchor points that are rough locations manually extracted from the 2D mine survey are used to further reduce the drift in the estimated trajectories. Leingartner et al. [66] evaluate how well off-the-shelf sensors and mapping solutions work in two different field experiments, one from a disaster-relief operation in a 1.5 km long motorway tunnel, and one from a mapping experiment in a partly closed down tunnel. The authors conclude that despite advances in visual SLAM systems, lidar-based solu-

tions have a superior localization and mapping performance in dark and perceptually-degraded underground tunnels.

Jacobson et al. [67] propose a monocular SLAM system for online localization and mapping in an underground mine with minimal human intervention. Their method leverages surveyed mine maps to increase mapping accuracy and enabling global localization. In order to reduce the drift in the estimated robot trajectory a vision-based loop closure detection method is used where each newly obtained frame is compared to all previously obtained frames to detect loop closures. Ebadi et al. [4] present a centralized multi-robot SLAM system where a team of robots rely on onboard lidar scanners to autonomously explore and map network of complex underground tunnels.

2.5 Multi-Robot SLAM Architectures

Multi-robot SLAM systems have been becoming increasingly more popular over the past few decades due to their potential advantages compared to single-robot systems. This includes higher performance and efficiency in performing spatially or temporally distributed tasks such as exploration large-scale environments, higher fault-tolerance, information redundancy, and scalability. As illustrated in Fig. 2.2, multi-robot collaborative SLAM systems adopt either a centralized or decentralized architecture. Fig. 2.2(a) shows a centralized architecture where all robots communicate with a base station, where the base station is usually in charge of receiving telemetry, observations and local maps from all robots to estimate the state and trajectory of each robot and to fuse the received maps into a globally consistent representation of the environment.

Fig. 2.2(b) illustrates a decentralized architecture, where each agent only communicates with nearby agents. The major challenge in distributed systems is that heavy computational tasks are distributed among agents with limited onboard memory and

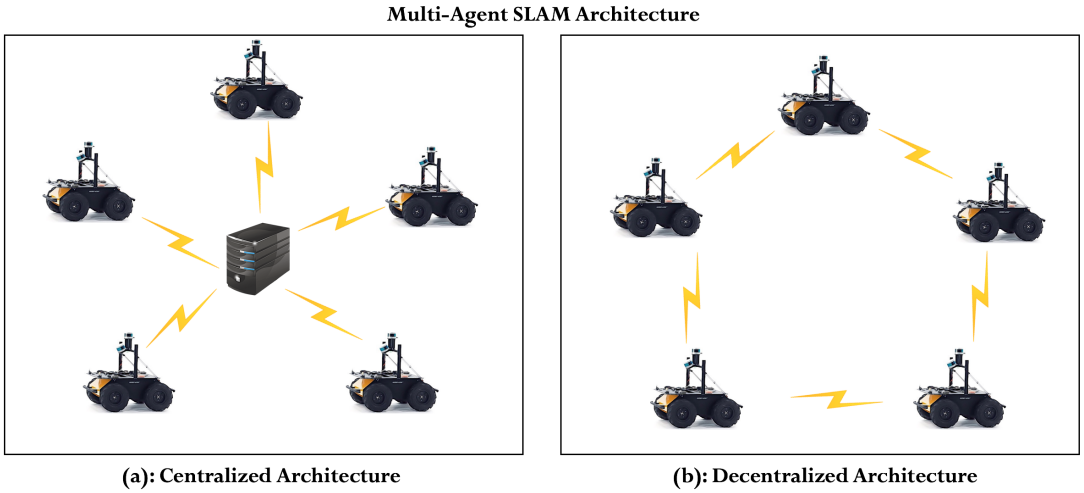

Fig. 2.2: Illustration of centralized and decentralized multi-robot systems.

computational resources.

While the SLAM community has made dramatic progress over the past decades in the field of single robot SLAM [57], extending these approaches to multi-robot systems in large-scale and ambiguous environments remains a challenge. One of the main underlying problems in multi-robot collaborative mapping is the map alignment and merging that aims to find the correspondences between the local maps obtained by multiple robots in order to consistently merge them into a global map. This can be viewed as a data association problem which manifests itself in image registration [68], and point cloud registration [69–71], in 2D and 3D spaces respectively.

A variety of multi-robot SLAM algorithms [48, 49, 53, 72–78] have been proposed that rely on 2D grid-based representation of the environment. Blanco et al. [73] present an approach based on extracting feature descriptors by means of a polar coordinate transformation around highly distinctive points in order to align occupancy grid maps which are discretized representations of 3D point clouds. In a similar study, Blanco et al. [75] present a new approach to matching occupancy grid maps by formulating the problem as a special instance of generic image registration and introduce a modified

RANSAC algorithm [79] to search for a dynamic number of internally consistent subsets of feature pairings to compute hypotheses about the translation and rotation between the maps.

Zhu et al. [76] formulated the grid map merging as the point set registration problem where the trimmed iterative closest point (TrICP) was used for map merging. Stefano et al. [72] propose a method that deals with low quality maps based on probabilistic grids. They address map merging using a motion planning algorithm where merging process is done by rotating and translating the partial maps until similar regions overlap.

A common approach to the map alignment problem is to interpret the input maps with an abstract representation. Kakuma et al. [80] propose a method using graph matching to align occupancy grid maps. They extract the graphs from the maps using region segmentation and obtain corresponding points using graph matching to estimate a similarity transformation matrix. Carlone et al. [74] propose an efficient extension of Rao-Blackwellized Particle Filters (RBPF) to the multi robot scenario in a grid-based map representation. The robots are required to exchange a small amount of information only when a rendezvous event occurs and to measure relative poses during the meeting. Their approach allows to jointly estimate the posterior of the robots by fusing the proprioceptive and the exteroceptive information exchanged among teammates.

2.6 Degeneracy and Loss of Observability

Accuracy of point cloud registration is an important factor in determining the reliability of lidar odometry and loop closure detection. As presented in Section 2.3.1, the ICP algorithm estimates the homogeneous transformation matrix consisting of a rotation and translation that best aligns two lidar scans. The reliability and accuracy of ICP-based scan matching is decreased in environments with sparse salient geometric structures and

features (e.g., corridors or tunnels with flat and symmetric walls). Thus, when mapping an unknown environment, it is crucial to determine if the local scene has sufficient geometric saliency to constrain the alignment between lidar scans.

2.6.1 Localization of salient features in 2D

Accurate detection and localization of two dimensional (2D) image features is crucial in vision tasks such as camera calibration, image matching, object tracking, or structure from motion. Harris corner detector introduced by Chris Harris et. al. [28] is a simple and popular technique for locating points with strong intensity variations in an image based on Taylor expansion of the directional intensity variation computed on an image point. Harris corner detector relies on the autocorrelation function of the image for measuring the intensity differences between a patch and windows shifted in different directions. Analysis of the eigenvalues of the autocorrelation matrix depends on the gradient of the image and the convolution with a window function that is either a rectangular window or Gaussian window.

The idea behind the Harris method is to detect salient points based on the difference in intensity for a displacement of (u, v) in all directions around a point in the image. This can be interpreted as finding windows that maximize the value of E as given by:

$$E(u,v) = \sum_{x,y \in w} \underbrace{w(x,y)}_{\text{window function}} \; [\underbrace{I(x+u, y+v)}_{\text{shifted intensity}} - \underbrace{I(x,y)}_{\text{intensity}}]^2, \tag{2.29}$$

where u is the window's displacement in the x direction, v is the window's displacement in the y direction, $E(u, v)$ is the change function defined as the sum of squared differences (SSD), $w(x, y)$ is the window centered at coordinate (x, y) in the image, ensuring that only the desired window is used. $I(x, y)$ is the intensity of the image at an image coordinate (x, y).

In order to find a corner point, E has to be maximized, which occurs when the shifted intensity function is maximized. For a small displacement u in the x direction, the shifted intensity function can be approximated by applying Taylor expansion as given by

$$I(x+u) \simeq I(x) + \nabla I(x)^T u, \tag{2.30}$$

where $\nabla I(x)$ is the gradient (vector of first-order partial derivatives) of $I(x)$.

For displacements of u and v in x and y directions respectively, the right hand of (2.29) becomes

$$I(x+u, y+v) \simeq I(x,y) + I_x(x,y)u \; I_y(x,y)v \tag{2.31}$$

$$= I(x,y) + [I_x(x,y)I_y(x,y)] \begin{bmatrix} u \\ v \end{bmatrix}, \tag{2.32}$$

where $I_x(.,.) = \frac{\partial I}{\partial x}$, and $I_y(.,.) = \frac{\partial I}{\partial y}$, denote partial derivatives in x and y.

The quadratic approximation of (2.29) simplifies to

$$E(u,v) \simeq \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.33}$$

where

$$M = \sum_{x,y \in w} w(x,y) \begin{bmatrix} \frac{\partial^2 I}{\partial x} & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \frac{\partial^2 I}{\partial y} \end{bmatrix} = \sum w(x,y) \begin{bmatrix} \frac{\partial I}{\partial x} \\ \frac{\partial I}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{bmatrix} = \sum w(x,y) \nabla I (\nabla I)^T, \tag{2.34}$$

is the second moment matrix (structure tensor) obtained from the sum of outer product

of image derivatives, that captures the intensity structure of the local neighborhood.

Let λ_1 and λ_2 be the eigenvalues obtained from the eigen decomposition of M as given by

$$M = R \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R^T. \tag{2.35}$$

When (2.29) is maximized, the SSD is large in shifts for all directions. By analyzing the eigenvectors of M, the directions for both the largest and smallest increases in SSD can be determined, where the corresponding eigenvalues correspond to the amount of increases in each direction. A score, r, is calculated for each window as given by

$$r = det(M) - k(trace(M))^2, \tag{2.36}$$

where $det(M) = \lambda_1 \lambda_2$ and $trace(M) = \lambda_1 + \lambda_2$.

As illustrated in Fig. 2.3 three cases can be considered:

1. If both λ_1 and λ_2 are small, then $|r|$ is going to be small, which indicates small variations in M, meaning the windowed image region is of approximately constant intensity.

2. If one eigenvalue is significantly larger than the other eigenvalue, then $r < 0$, which indicates local shifts in one direction cause small variations in M, and large variations in the orthogonal direction, indicating the windowed image region is ridge shaped (i.e., is an edge).

3. If both eigenvalues are large, then r is large, indicating the shifts in any direction will result in a significant increase, meaning the windowed image region is a corner whose local neighborhood stands in two dominant and different edge directions.

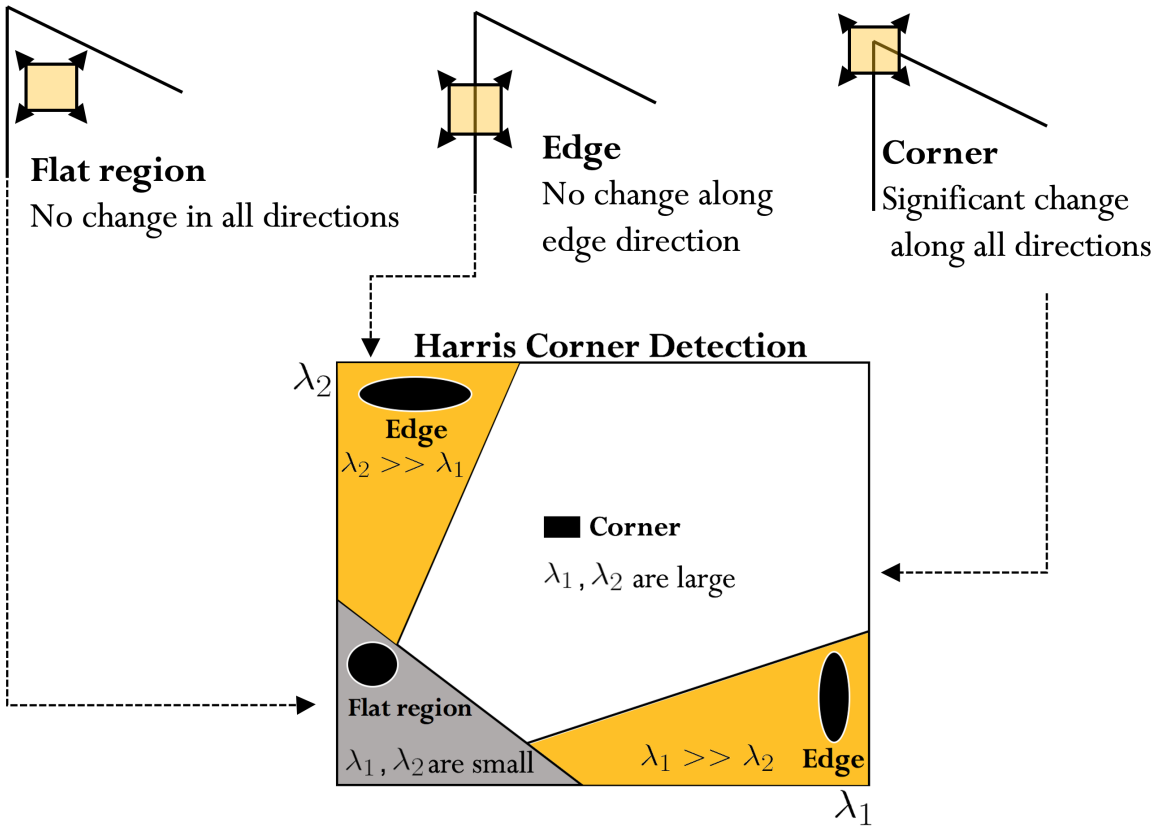

Fig. 2.3: Harris corner detection - An illustration of an edge and corner detection from eigenvalue analysis of the structure tensor.

As presented in Chapter 4, this thesis extends this method to three dimensional spaces, by looking at the gradients of a squared error cost function to determine the presence of salient geometric structures in unknown 3D environments.

2.6.2 Localization of salient features in 3D

While recent developments on lidar-based state estimation have shown promising results in localization and mapping with low drift over long distances [4], the problem of real-time determination of geometric saliency is an important and less studied problem. Determination of level of geometric degeneracy in an unknown environment enables the analysis of reliability of lidar scan registration. The reliability of lidar-based odometric

estimates is reduced by the noise and degeneracy that can arise from symmetry and self-similarity of the environment [81]. As illustrated in Fig. 2.4, a tunnel with planar and featureless walls is an example of an environment where the scene geometry does not constrain the ICP-optimization sufficiently [82]. In this environment, the motion of the robot along the direction of the tunnel becomes unobservable as lidar scan matching becomes a highly non-convex optimization problem with many extrema which leads to inaccurate estimations of 3D motion. This leads to significant drift in the estimated trajectory of the robot in long-term or large-scale navigation.

Over the past decade, there has been a number of efforts to evaluate the reliability of ICP scan registration by estimating the covariance of the ICP algorithm [81, 83–86]. Andrea Censi [81], proposed a method based on the analysis of 2D point-to-plane error function being minimized, where the covariance of the ICP algorithm is defined as the statistical dispersion due to three main sources of error, namely wrong convergence, underconstrained situations, and sensor noise. Manoj et al. [83] extends [81] by presenting a closed-form solution to estimate the covariance of the resultant transformation provided by the ICP algorithm for 3D point cloud registration using a point-to-point error function.

Brossard et al. [84] propose a novel approach to 3D ICP covariance computation that in

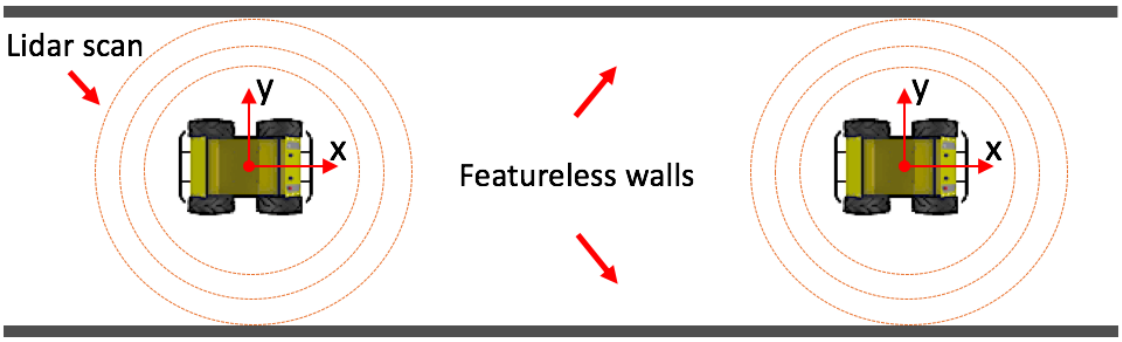

Fig. 2.4: An illustration of a geometrically degenerate scene in a featureless and symmetrical tunnel.

addition to accounting for all the sources of errors as listed in Censi's work [81], it also accounts for the sensor bias noise. Authors show the observed points in two point clouds share common errors that stem from stability of the laser, observed material, and beam angles or wrong intrinsic calibration process that could strictly limit the confidence in ICP estimates.

The covariance of ICP estimates not only depends on sensor noise characteristics, or convergence of the algorithm but also on the geometry of the environment. In environments with insufficient salient geometric features the ICP may become trapped in a local minima and produce poor estimations of motion between lidar scans. Zhang et. al, [82] proposes a method to mitigate the degeneracy in optimization-based problems through analysis of geometric structure of the problem constraints. Given a linearized system as

$$\underset{x}{\operatorname{argmin}} ||Ax - b||^2 \tag{2.37}$$

authors show the lack of observability in the environment manifests itself in pose graph optimization, as the linear system to be solved in iterative methods becomes rank-deficient. This results in a degraded optimization problem, where some directions of the state space are loosely constrained. After determining the eigenvalues and eigenvectors from the A matrix of the linearized system, the authors compare the individual eigenvalues to a threshold to determine which directions are degenerate. In order to find the appropriate threshold, a sample dataset with predetermined degenerate and nondegenerate sections is used. While this method performs well in determining loosely constrained directions of the state space, the reliance on a training dataset makes it an unsuitable choice for real-time applications or applications in unknown environments where no training dataset is available for finding the thresholds.

Rong et. al. [87] propose the formulation of an online methodology to detect future motions that can result in degraded perceptual information leading to state estimation inaccuracy and degeneracy in a monocular visual-inertial state estimation system. The monocular visual-inertial state estimation is formulated in the standard state-equation form as given by

$$x[k+1] = \mathbf{A}x[k] + \mathbf{B}u[k] \tag{2.38}$$

$$Y[k] = \mathbf{x}[k] + \mathbf{D}u[k]. \tag{2.39}$$

The observability of the LTI system can be determined by looking at the pair $(\mathbf{A}, \mathbf{C})$ through analysis of the eigenvalues of the empirical observability gramian (EOG) as given by

$$W_{do} = \sum_{m=0}^{\infty} (A^T)^m C^T C A^m, \tag{2.40}$$

where W_{do} is the Discrete Observability Gramian that quantifies the degree of system parameter observability given sensor measurements and serves as an online numerical observability analysis tool to directly identify the condition degeneracy. The eigenvalues of W_{do} represent the relative observability of each system parameter, where the condition number defines the relative scaling between the most and least observable parameters. A large condition number indicates that the observability is dominated by a subset of parameters while the other parameters are weakly observed.

Nobili et al [88] propose a method to predict the risk of failed alignment between two point clouds as a function of the overlap between two point clouds and the constraints available in the region of overlap between them. Authors adopt a data-driven approach to evaluate the geometric constraints available for alignment and the volume of spatial overlap between the clouds, and use this data to learn a model to predict the risk of a

failed alignment between two lidar scans.

In nonlinear optimization methods such as the gradient descent or steepest descent methods where the objective is to minimize a given cost function, the gradient vector at each point is used as the search direction for each iteration of the algorithm, where the gradient vector at a point, provides the direction of maximum rate of change (maximum increase) of the function at that point. The second derivative of the cost function at a point measures the instantaneous rate of change of the first derivative and indicates whether a gradient step drops the cost as much as the gradient alone may imply.

For a scalar-valued cost function with multiple variables, a Hessian encodes the second derivatives of a cost function with respect to all pairs of variables. The analysis of the eigenvalues of the Hessian is a useful tool to determine the rate of convergence of the cost function. The maximum eigenvalue λ_{max} of the Hessian corresponds to the maximum second derivative or the maximum convergence rate, whereas, the minimum eigenvalue λ_{min} corresponds to the smallest second derivative or the slowest convergence rate of the cost function. Gradient descent methods work poorly if the gradients in different directions are in different order of magnitude.

In an ideal case where $\lambda_{max} = \lambda_{min}$, the basins of the cost function would be perfectly circular. In this scenario, an optimization method such as the steepest descent could converge to the minimum in a single iteration, given an appropriate step length. In contrast, in the case where λ_{min} is a very small number, while convergence could still be achieved but it may take an infinite number of iterations.

The ratio of the largest eigenvalue to the smallest eigenvalue, $\lambda_{max}/\lambda_{min}$, is known as the condition number. The Hessian is ill-conditioned if the condition number is very high, usually indicating that the λ_{min} is orders of magnitude smaller than the λ_{max}. In this case, the gradient along the eigenvector with the largest eigenvalue is much smaller

than the one with the smallest eigenvalue, meaning that the basins of the cost function have contours that are very long *ellipsoids* rather than being close to *circular*. This means, the gradient descent methods will either learn too slowly in the low gradient direction and/or overshoot the solution in the high gradient direction.

Many optimization methods, such as Newton's method, require the computation of the inverse of the Hessian. The singular value decomposition of the Hessian matrix is given by $H = U\Sigma V^T$, where U and V are the left-singular vectors and right-singular vectors of H respectively, and Σ is a diagonal matrix where its diagonal entries are known as the singular values of H. Assuming H is invertible, the inverse can be computed as $H^{-1} = V\Sigma^{+}U^T$. If the Hessian is ill-conditioned, its inverse can be numerically unstable since the smallest singular value will become a very large number, meaning that any noise in computing the Hessian amplifies dramatically when the Hessian is inverted. In this dissertation, metrics are developed using the eigenvalue analysis of the Hessian of a distance cost function based on the rigid 3D transformation obtained from the ICP algorithm to evaluate the level of geometric degeneracy in an unknown environment.

Chapter 3

Lidar Odometry and Mapping

3.1 Overview

The precise estimation of 3D position and orientation of a mobile robot is crucial when performing tasks like trajectory planning, manipulation and mapping. A robot exploring an unknown and GPS-denied environment relies on onboard sensors such as lidar, camera, and Inertial Measurement Unit (IMU) to estimate its motion in real-time. Lidar has been a popular solution to ego-motion estimation and 3D mapping, especially in dark and perceptually-degraded environments, as it provides high resolution range data over a 360° horizontal field of view at a high temporal and spatial sampling rate, and does not rely on external light sources. In lidar odometry, the robot trajectory is computed by incrementally estimating the state of the robot through examination of the changes that motion induces on the obtained consecutive lidar scans and finding the most probable relative 3D motion between consecutive lidar scans.

As presented in Fig. 3.1, a lidar SLAM front-end is comprised of three key components: (i) scan matching and odometry, (ii) loop closure detection, and (iii) 3D map

construction. In the rest of this section, detailed description of each layer is provided.

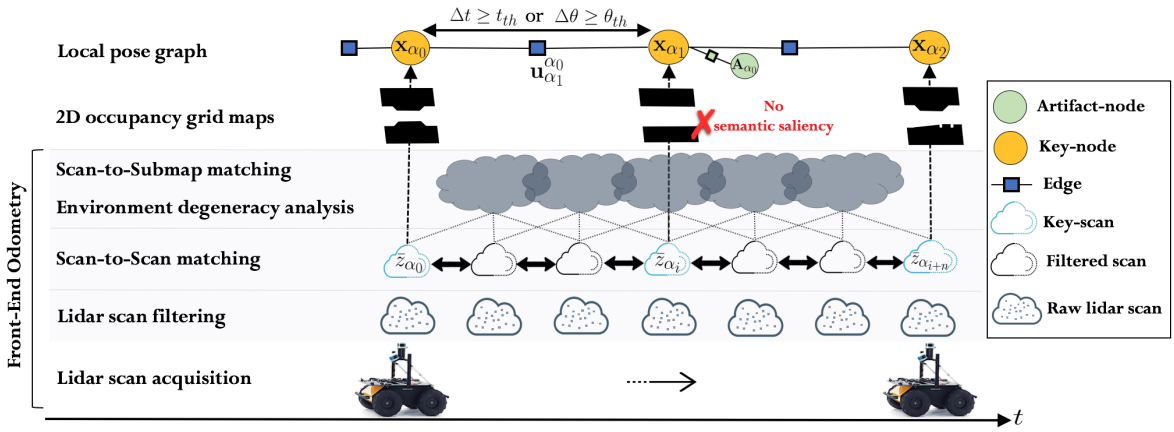

Fig. 3.1: Overview of the lidar-based front-end and the local pose graph.

3.2 Lidar Point Cloud Filtering

A 3D lidar scanner collects a series of points with precise angle and distance information. The raw point clouds obtained in each lidar scan are often density-uneven and contain a massive amount of redundant and noisy points. Thus, using raw point clouds not only can be computationally expensive, but can also lead to noisy odometric estimates. In ICP and its variants, finding the nearest neighbors and determining the correspondences is the most computationally expensive step, since for every point in the source point cloud, the closest point in the reference point cloud needs to be determined. In order to reduce the noise and redundant information, and improve the scan registration performance, most methods use a subset of the raw point cloud data [89].

Among different point cloud filtering methods, random downsampling, and voxel grid filtering [32, 89–91] are commonly used to downsample raw point cloud data. In the random downsampling method, points are randomly sampled with a uniform probability, whereas, as illustrated in Fig. 3.2 in voxel grid downsampling, a 3D voxel grid is created

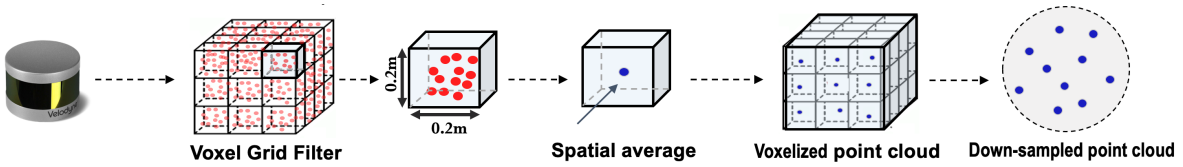

Fig. 3.2: Voxel-grid filtering

over the input point cloud data and the points in each voxel are approximated with their centroid value. While these filtering methods are effective in downsampling a raw point cloud data, they remove points from a lidar scan regardless of their importance and effect on the performance and accuracy of lidar odometry. Feature-based downsampling methods instead rely on detection, and extraction of salient geometric features such as planar and edge features [12], ellipsoidal surfel features [7] and ground features [6]. In this thesis, using the method of Zhang et al. [12], up to 90% of the points in every raw lidar scan are decimated by extracting a set of uniformly distributed salient features located on sharp edges and planar surfaces.

As illustrated in Fig. 3.3, a lidar scanner such as Velodyne Puck Lite VLP16 has 16 laser channels or scan planes. A laser channel is a single 903 nm laser emitter and detector pair. Each laser channel is fixed at a particular elevation angle relative to the horizontal plane of the sensor. Using the LOAM feature extraction method, the smoothness of the local surface for each scan plane is evaluated separately and the points in the scan plane are sorted based on their smoothness values. In order to obtain uniform distribution of features in a scan, each scan plane is divided into four identical sub-regions where a fixed number of edge and planar features are extracted from each sub-region. Points

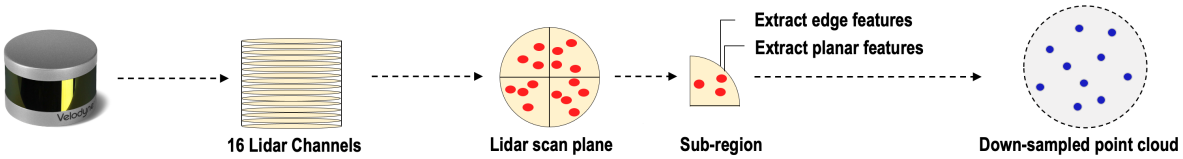

Fig. 3.3: Lidar point cloud downsampling using LOAM feature extraction method.

with maximum smoothness values are selected as planar features, whereas, points with minimum smoothness values are selected as edge features.

3.3 Lidar Odometry

As a robot navigates the environment and obtains lidar scans, the SLAM front-end relies on the ICP algorithm to compute the 3D motion of the robot between the consecutive lidar scans. As this thesis addresses the problem of localization and mapping in extreme and remote environments, the scene is assumed to be static. Furthermore, it is assumed that the lidar operates at a high temporal sampling rate relative to robot motion, such that distortions in the point cloud due to the motion of the robot are negligible and can be ignored. In order to estimate the motion of the robot between consecutive lidar scans, a two-stage scan registration process is used, namely scan-to-scan and scan-to-submap matching. In the scan-to-scan registration step, the incoming point cloud is aligned to the last obtained lidar scan to find a rigid transformation that best aligns the consecutive point clouds. After this approximate localization, the solution is further refined by registering the incoming point cloud to a submap created from the local region of previously collected point cloud data in the pose graph. Given the current estimated pose of the robot in the world coordinate system $\mathcal{W}$, the incoming lidar scan is transformed to the world coordinate system using the same transformation, and an approximate nearest neighbors search is performed to find the closest points in the map. A second point cloud registration similar to scan-to-scan registration is used to obtain the odometric estimate $\mathbf{u}_{t+1}^{t} \doteq [\mathbf{R}_{t+1}^{t}, \mathbf{t}_{t+1}^{t}] \in SE(3)$ that describes the 3D motion of the robot between times t and $t+1$.

3.3.1 Scan-to-Scan Matching

Let $\bar{z}_t$ and $\bar{z}_{t+1}$ denote two consecutive filtered lidar scans. Since the process happens on the same robot, the robot name is dropped for brevity. In the scan-to-scan matching step, using an initial alignment estimate of identity rotation and zero translation, the nearest neighbor search is used to find the set of corresponding points $\check{P}^t \subset \bar{z}_t$ and $\check{P}^{t+1} \subset \bar{z}_{t+1}$, and a new 3D rigid body transformation $\hat{\mathbf{u}}_{t+1}^t \doteq [\mathbf{R}_{t+1}^t, \mathbf{t}_{t+1}^t] \in SE(3)$, that best aligns the set of corresponding points to update the initial estimate. These steps are repeated until convergence is achieved or a maximum number of iterations has occurred. We define the residual error between a pair of corresponding points by

$$\mathbf{d}_k = \hat{\mathbf{u}}_{t+1}^t \check{p}_k^t - \check{p}_k^{t+1}, \tag{3.1}$$

where $\mathbf{d}_k = [d_{k_x}, d_{k_y}, d_{k_z}]$ is the misalignment vector between the kth pair of corresponding points. The mean squared distance between the aligned point clouds is given by

$$\mathcal{E}(\hat{\mathbf{u}}_{t+1}^t \check{P}^t, \check{P}^{t+1}) = \frac{1}{N_k} \sum_{k=1}^{N_k} \|\mathbf{d}_k\|^2, \tag{3.2}$$

where $\mathcal{E}(.)$ is a scalar valued cost function that computes the mean squared error, N_k is the total number of corresponding points, and $\|.\|$ denotes the $\mathcal{L}_2$ norm. The scan matching can be formulated as an optimization problem that minimizes (3.2) as given by:

$$\hat{\mathbf{u}}_{t+1}^t = \underset{\mathbf{u}_{t+1}^t}{\operatorname{argmin}}\, \mathcal{E}(\mathbf{u}_{t+1}^t \check{P}^t, \check{P}^{t+1}). \tag{3.3}$$

and it can be implemented as given by Eq. (2.3) using the process detailed in Section 2.3.1.

3.3.2 Scan-to-Submap Matching

In order to further refine the odometric estimate obtained from solution of scan-to-scan matching (3.3), in this thesis a scan-to-submap matching step is used by using a submap created from the local region of previously collected point cloud data in the pose graph. Given the current estimated pose of the robot in the world coordinate system $\mathcal{W}$, the incoming filtered lidar scan is first transformed to the world coordinate system and an approximate nearest neighbors search is performed to find the closest points in neighboring key-scans. After finding the nearest neighbor points and forming a submap, similar to (3.3), an ICP-based scan registration is performed to obtain the refined odometric measurement $\overset{*}{\mathbf{u}}_{t+1}^{t}$. While both scan-to-scan and scan-to-submap matching steps produce odometric measurements, as shown in the real-world experiments in this section, the scan-to-submap matching step improves the accuracy of motion estimates due to better initial alignment of the point clouds that improve the convergence of the ICP algorithm to the optimal solution.

3.3.3 Pose Graph Construction

As the lidar operates at a high temporal sampling rate relative to robot motion, the graph would be over-sampled in time if a new node were instantiated after each new odometric measurement. A variety of pose graph sparsification methods [92–94] have been proposed to address scalability and computational load associated with pose graph optimization. Pose graph sparsification addresses scalability by reducing the number of nodes added to the graph, or by pruning less informative nodes and factors [57]. In this work, we utilize a reduced pose graph where a new node is instantiated in the graph after reaching a minimum odometric displacement (30° rotation or 1 m translation). We refer to the nodes in the reduced graph as *key-nodes* and lidar scans associated with

key-nodes as *key-scans*. The relative 3D motion $\mathbf{u}_{i+1}^{i}$ between key-nodes i and $i+1$ in the pose graph is obtained from integration of estimated motions between times t and $t+n_i$ and the 3D map of the environment can be obtained from the accumulation of all point clouds in the key-scans.

3.4 Experimental Results and Analysis

The performance of front-end odometry is evaluated for both scan-to-scan and scan-to-submap matching steps by measuring the drift in robot trajectory for a variety of trajectory lengths. Fig. 3.4 reports the accuracy of lidar odometry by measuring the Relative Pose Error (RPE) as a function of distance traversed using the package for evaluation of odometry and SLAM (EVO) [95] in two underground mines. In the Safety Research mine, the robot autonomously navigates 1400m of the mine tunnels. By relying on scan-to-scan registration, the accumulation of errors results in more than 18m (6% relative position error) average drift in translation for each 300m of traversed distance. By using the scan-to-submap registration step the RPE is reduced to less than

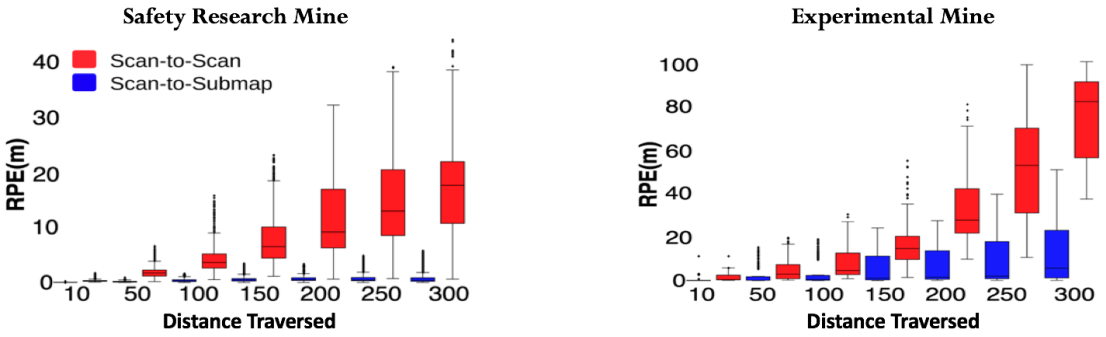

Fig. 3.4: Quantitative evaluation of drift using scan-to-scan and scan-to-submap matching as a function of distance traversed in Bruceton Safety Research and Experimental mines. Each box comprises the RPE values ranging from the first to the third quartile. The median is indicated by the black horizontal bar. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. Outliers are shown as dots.

1% of distance traversed for each 300m of travel.

The Experimental mine presents a more challenging subterranean environment with long and featureless corridors. During the 700m autonomous navigation, due to high level of geometric degeneracy at multiple locations, accumulation of errors from scan-to-scan registration leads to more than 80m (27% relative position error) of average drift in translation for each 300m of distance traversed. By using the scan-to-submap registration step, the estimated motion of the robot is further refined and the drift in translation is reduced to 7m (2% relative position error) of average drift for each 300m of traversed distance. Fig. 3.5 presents partial trajectories in the Safety Research and Experimental mines using scan-to-scan and scan-to-submap matching methods. The results show scan-to-scan matching poorly estimates the angular and translational motion of the robot which leads to significant drift in the estimated robot trajectory.

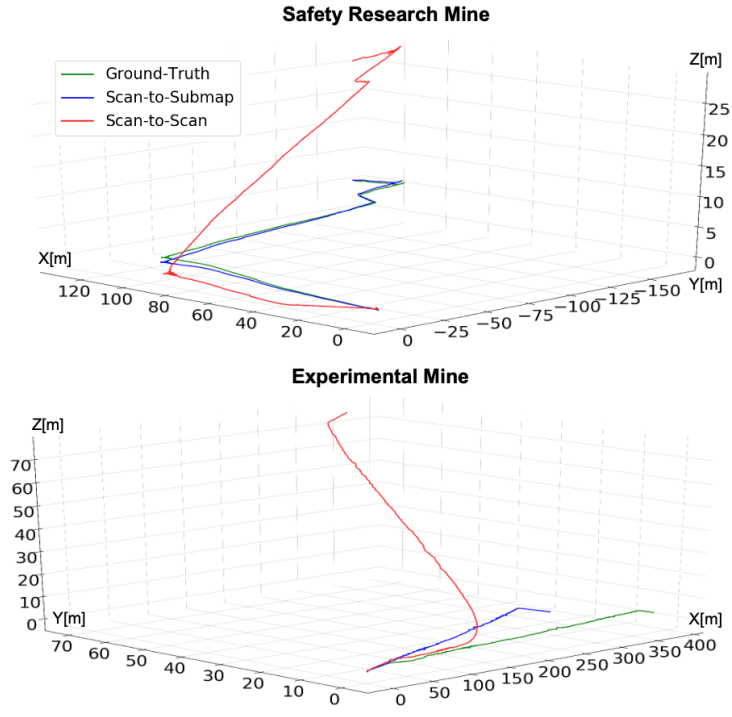

Fig. 3.5: Robot trajectories obtained using scan-to-scan and scan-to-submap registration methods.

Chapter 4

Real-time Determination of Geometric Degeneracy

Detection of loss of observability in an unknown environment is a crucial capability to evaluate the reliability of a SLAM front-end in reconstructing its full internal state based on observations of the environment. Loss of observability can occur in any odometry system; scarcity of texture and salient features in a vision-based system, lack of thermal variations in a thermally flat environment (e.g., cold underground mines and caves) in a thermal-inertial odometry, lack of distinctive geometric structures in a lidar-based SLAM front-end, or slippery terrains in a wheel-inertial odometry system. As presented in Section 2.6.2, loss of observability may occur when a robot traverses a tunnel or corridor with long, symmetrical and featureless walls.

This real-time determination of geometric degeneracy is crucial in improving intra- and inter-robot loop closure detections; by determining the unobservable areas in an unknown environment, the front-end can constrain the search for loop closures to only areas with sufficient geometric observability where lidar scan matching is well-constrained and the ICP algorithm has a higher probability of converging to the optimal solution.

This fundamental capability can reduce the number of spurious intra- and inter-robot loop closures in both single and multi-robot SLAM systems. In this chapter, a method is developed to determine the level of geometric degeneracy in an unknown environment in real-time using a cost function based on the estimated 3D transformation obtained from the ICP-based lidar scan matching.

4.1 Analytical Study of Bias and Uncertainty in Lidar Scan Registration

In this section, analytic computation of bias and uncertainty in the optimal 3D transformation between a pair of lidar scans is presented. Fig. 4.1 illustrates a robot equipped with a 3D lidar scanner exploring an unknown environment. The robot obtains two lidar scans P^i and Q^j at poses i and j respectively. Denoting the true position of the kth observed landmark (the red star in Fig. 4.1) in the environment relative to robot poses i and j with $\mathbf{x}_k^i$ and $\mathbf{x}_k^j$, the lidar points corresponding to the observed landmark

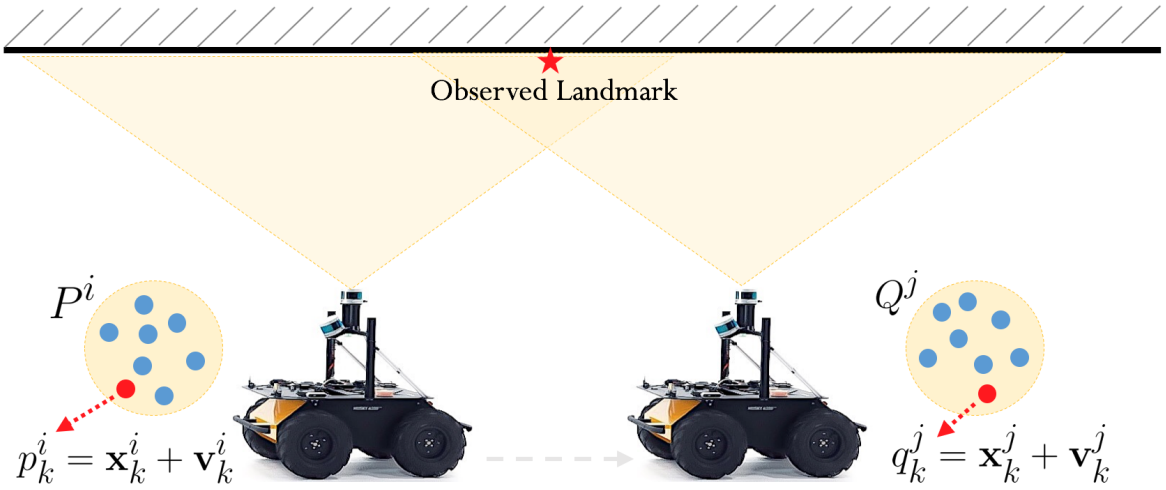

Fig. 4.1: (a) A robot obtains two lidar scan measurements at poses i and j in the environment.

can be defined as

$$p_k^i = \mathbf{x}_k^i + \mathbf{v}_k^i, \tag{4.1}$$

$$q_k^i = \mathbf{x}_k^j + \mathbf{v}_k^j, \tag{4.2}$$

$$\mathbf{v}_k^i, \mathbf{v}_k^j \sim \mathcal{N}(0, \sigma_m^2) \tag{4.3}$$

where $\mathbf{v}_k^i, \mathbf{v}_k^j$ are independent lidar measurement noises at robot poses i and j, assumed to be sampled from a Gaussian distribution with zero mean, and variance σ_m^2.

Consider the simple case where the robot motion between poses i and j consists of only translation and no rotation. The true relative position of all landmarks in P^i and Q^j can be written in terms of the true landmark positions observed in robot pose i and the true translation of the robot from pose i to j as given by

$$\mathbf{x}_k^j = \mathbf{x}_k^i + \overset{*}{\mathbf{t}} \quad \text{for} \quad k = 1 : N_k, \tag{4.4}$$

where $\overset{*}{\mathbf{t}} \doteq [\overset{*}{t}_x, \overset{*}{t}_y, \overset{*}{t}_z]$ is the true 3D translation vector between robot poses i and j. The robot pose indices are dropped from superscript for brevity.

Using the ICP algorithm to align the lidar scans, let $\mathbf{t}$ denote the optimal translation between the point clouds P^i and Q^j as given by

$$\mathbf{t} \doteq [t_x, t_y, t_z]. \tag{4.5}$$

The residual distance between the kth pair of aligned points is given by

$$d_k = p_k^i + \mathbf{t} - q_k^j, \tag{4.6}$$

where p_k^i and q_k^i denote the kth pair of corresponding points, and d_k is the residual distance. Assuming perfect correspondences such that p_k^i and q_k^i correspond to the same exact landmark in the environment we can substitute from (4.1) and (4.2) into (4.6):

$$d_k = (\mathbf{x}_k^i + \mathbf{v}_k^i) + \mathbf{t} - (\mathbf{x}_k^j + \mathbf{v}_k^j) \tag{4.7}$$

$$= \underbrace{(\mathbf{x}_k^i + \mathbf{t} - \mathbf{x}_k^j)}_{\mathbf{c}_k} + \underbrace{(\mathbf{v}_k^i - \mathbf{v}_k^j)}_{\mathbf{m}_k}, \tag{4.8}$$

where $\mathbf{m}_k$ is the error due to lidar measurement noise, and $\mathbf{c}_k$ is the correspondence error that consists of two factors; the incorrect computation of translation $\mathbf{t}$, and the mismatch between aligned points such that $\mathbf{x}_k^i$ and $\mathbf{x}_k^j$ do not correspond to the same landmark in the environment. In the case of perfect correspondences, we can substitute (4.4) into (4.8):

$$d_k = (\mathbf{x}_k^i + \mathbf{t}_k - \mathbf{x}_k^i - \overset{*}{\mathbf{t}}_k) + (\mathbf{v}_k^i - \mathbf{v}_k^j) \tag{4.9}$$

$$d_k = (\mathbf{t} - \overset{*}{\mathbf{t}}) + (\mathbf{v}_k^i - \mathbf{v}_k^j). \tag{4.10}$$

Let $\xi(\mathbf{t})$ denote a scalar-valued cost function for the set of all corresponding points, defined as the sum of the squared distances between the aligned points as given by

$$\xi(\mathbf{t}) = \sum_{k=1}^{N_k} ((d_{k_x})^2 + (d_{k_y})^2 + (d_{k_z})^2). \tag{4.11}$$

To find the optimal translation $\mathbf{t}$ that minimizes (4.11), the gradient of the cost function

is computed as given by

$$J = \frac{\partial \xi(\mathbf{t})}{\partial (t_x, t_y, t_z)} = \begin{bmatrix} \frac{\partial \xi(\mathbf{t})}{\partial t_x} & \frac{\partial \xi(\mathbf{t})}{\partial t_y} & \frac{\partial \xi(\mathbf{t})}{\partial t_z} \end{bmatrix}^T \tag{4.12}$$

$$= \begin{bmatrix} 2\sum_{k=1}^{N_k} d_{k_x} & 2\sum_{k=1}^{N_k} d_{k_y} & 2\sum_{k=1}^{N_k} d_{k_z} \end{bmatrix}^T, \tag{4.13}$$

and solve the set of equations by setting the gradient to zero. As no rotation is involved in this scenario, the three parameters of the translation vector are separable and the optimal values of three components of the translation vector can be found by setting each partial derivative to zero. The optimal solution for t_x is given by

$$2\sum_{k=1}^{N_k} d_{k_x} = 2\sum_{k=1}^{N_k}((t_x - \overset{*}{t}_x) + (\mathbf{v}_{k_x}^i - \mathbf{v}_{k_x}^j)) = 0 \tag{4.14}$$

$$= 2N_k(t_x - \overset{*}{t}_x) + 2\sum_{k=1}^{N_k}(\mathbf{v}_{k_x}^i - \mathbf{v}_{k_x}^j) = 0 \tag{4.15}$$

$$t_x = \overset{*}{t}_x + \frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}_{k_x}^j - \mathbf{v}_{k_x}^i) \tag{4.16}$$

Similarly the optimal solutions for t_y and t_z are given by

$$t_y = \overset{*}{t}_y + \frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}_{k_y}^j - \mathbf{v}_{k_y}^i), \tag{4.17}$$

$$t_z = \overset{*}{t}_z + \frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}_{k_z}^j - \mathbf{v}_{k_z}^i). \tag{4.18}$$

To assess the bias in the estimated optimal transformation between two point clouds

the expected value of each component of the 3D translation is evaluated as given by

$$\mathbb{E}[t_x] = \mathbb{E}[\overset{*}{t}_x] + \mathbb{E}[\frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}^j_{k_x} - \mathbf{v}^i_{k_x})] \tag{4.19}$$

$$= \overset{*}{t}_x + \frac{1}{N_k}\sum_{k=1}^{N_k}\mathbb{E}[\mathbf{v}^j_{k_x} - \mathbf{v}^i_{k_x}]. \tag{4.20}$$

Since the added noise is assumed to be zero mean

$$\mathbb{E}[t_x] = \overset{*}{t}_x + \frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbb{E}[\mathbf{v}^j_{k_x}] - \mathbb{E}[\mathbf{v}^i_{k_x}]) = \overset{*}{t}_x. \tag{4.21}$$

Thus, the expected value of computed translation along the x axis is equal to the true translation of the lidar along the x axis in the environment, indicating the optimal translation t_x is *unbiased*.

To evaluate the precision of the computed optimal translation, the variance of the error can be evaluated as given by

$$\mathrm{Var}(t_x - \overset{*}{t}_x) = \mathbb{E}[(t_x - \overset{*}{t}_x)^2], \tag{4.22}$$

where Var(.) denotes the variance. From (4.16) the error term can be written in terms of the measurement noises such that

$$t_x - \overset{*}{t}_x = \frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}^j_{k_x} - \mathbf{v}^i_{k_x}). \tag{4.23}$$

Substituting (4.21) and (4.23) into (4.22)

$$Var(t_x - \overset{*}{t}_x) = \mathbb{E}[(t_x - \overset{*}{t}_x)^2] \tag{4.24}$$

$$= \mathbb{E}[(\frac{1}{N_k}\sum_{k=1}^{N_k}(\mathbf{v}_{k_x}^j - \mathbf{v}_{k_x}^i))^2] = \frac{1}{N_k^2}\sum_{k=1}^{N_k}\mathbb{E}[(\mathbf{v}_{k_x}^j - \mathbf{v}_{k_x}^i)^2]$$

$$= \frac{1}{N_k^2}\sum_{k=1}^{N_k}\mathbb{E}[(\mathbf{v}_{k_x}^j)^2 + (\mathbf{v}_{k_x}^i)^2 - 2\mathbf{v}_{k_x}^i\mathbf{v}_{k_x}^j] \tag{4.25}$$

$$= \frac{1}{N_k^2}\sum_{k=1}^{N_k}(\mathbb{E}[(\mathbf{v}_{k_x}^i)^2] + \mathbb{E}[(\mathbf{v}_{k_x}^j)^2] - 2\mathbb{E}[\mathbf{v}_{k_x}^i\mathbf{v}_{k_x}^j]). \tag{4.26}$$

Using (4.3), $\mathbb{E}[\mathbf{v}_{k_x}^i, \mathbf{v}_{k_x}^j] = \sigma_m^2\delta(i-j)$, (4.26) can be written as

$$\mathbb{E}[(t_x - \overset{*}{t}_x)^2] = \frac{1}{N_k^2}\sum_{k=1}^{N_k}(\sigma_m^2 + \sigma_m^2) \tag{4.27}$$

$$= \frac{2N_k}{N_k^2}\sigma_m^2 = 2\sigma_m^2 N_k^{-1}. \tag{4.28}$$

Similarly for the y and z components of the optimal translation, the expected value and the variance of the error is given by

$$\mathbb{E}[t_y] = \overset{*}{t}_y, \tag{4.29}$$

$$\mathbb{E}[(t_y - \overset{*}{t}_y)^2] = 2\sigma_m^2 N_k^{-1} \tag{4.30}$$

and

$$\mathbb{E}[t_z] = \overset{*}{t}_z, \tag{4.31}$$

$$\mathbb{E}[(t_z - \overset{*}{t}_z)^2] = 2\sigma_m^2 N_k^{-1} \tag{4.32}$$

Observing that the variance of the error is inversely proportional to the number correspondences between the two point clouds highlights the importance of having a large

number of corresponding points in obtaining more accurate motion estimations between pairs of lidar scans. For a small number of corresponding points, even though the expected value of $\mathbf{t}$ is unbiased, it is likely to have high variability in the results. As the number of corresponding points is increased, there is a higher probability of being close to the expected value.

Similar to the method used in the Harris corner detector presented in Section 2.6.1, an approximation of the Hessian [96] of the cost function $\xi(\mathbf{t})$ is obtained by computing the sum of outer products of the gradient vectors defined in (4.12) computed for each pair of corresponding points:

$$H \simeq \sum_{k=1}^{N_k} J_k J_k^T, \tag{4.33}$$

where H is the approximated 3×3 Hessian matrix, where the second order terms are neglected as it is assumed around the minimum of the cost function the rate of change of the slope is minimal and the function is only mildly nonlinear.

For the special case where the 3D motion between two point clouds consists of only translation, the approximated Hessian of the cost function can be written as

$$H \simeq \sum_{k=1}^{N_k} J_k J_k^T = \sum_{k=1}^{N_k} \begin{bmatrix} 2d_{k_x} \\ 2d_{k_y} \\ 2d_{k_z} \end{bmatrix} \begin{bmatrix} 2d_{k_x} & 2d_{k_y} & 2d_{k_z} \end{bmatrix} \tag{4.34}$$

$$H \simeq \begin{bmatrix} \sum_{k=1}^{N_k}(2d_{k_x})^2 & \sum_{k=1}^{N_k}(2d_{k_x})(2d_{k_y}) & \sum_{k=1}^{N_k}(2d_{k_x})(2d_{k_z}) \\ \sum_{k=1}^{N_k}(2d_{k_y})(2d_{k_x}) & \sum_{k=1}^{N_k}(2d_{k_y})^2 & \sum_{k=1}^{N_k}(2d_{k_y})(2d_{k_z}) \\ \sum_{k=1}^{N_k}(2d_{k_z})(2d_{k_x}) & \sum_{k=1}^{N_k}(2d_{k_z})(2d_{k_y}) & \sum_{k=1}^{N_k}(2d_{k_z})^2 \end{bmatrix} \tag{4.35}$$

The distribution of the gradient vectors for the set of N_k corresponding points can be

determined by computing the expected value of the approximated Hessian as given by

$$\mathbb{E}[H] = \begin{bmatrix} \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_x})^2] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_x})(2d_{k_y})] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_x})(2d_{k_z})] \\ \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_y})(2d_{k_x})] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_y})^2] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_y})(2d_{k_z})] \\ \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_z})(2d_{k_x})] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_z})(2d_{k_y})] & \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_z})^2] \end{bmatrix} \tag{4.36}$$

The expected value of the first diagonal entry is given by

$$\mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_x})^2] = 4\sum_{k=1}^{N_k}\mathbb{E}[((t_x - \overset{*}{t}_x) + (\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x}))^2] \tag{4.37}$$

substituting (4.23) into (4.37)

$$\mathbb{E}[(2d_{k_x})^2] = 4\sum_{k=1}^{N_k}\mathbb{E}[(\frac{1}{N_k}\sum_{k'=1}^{N_k}(\mathbf{v}^j_{k'_x} - \mathbf{v}^i_{k'_x}) + (\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x}))^2] \tag{4.38}$$

$$= 4\sum_{k=1}^{N_k}\mathbb{E}[(\frac{1}{N_k}\sum_{k'=1}^{N_k-1}(\mathbf{v}^j_{k'_x} - \mathbf{v}^i_{k'_x}) + (\frac{-1}{N_k}(\mathbf{v}^i_{k'_x} - \mathbf{v}^j_{k'_x}) + \mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x}))^2] \tag{4.39}$$

$$= 4\sum_{k=1}^{N_k}\mathbb{E}[(\frac{1}{N_k}\sum_{k'=1}^{N_k-1}(\mathbf{v}^j_{k'_x} - \mathbf{v}^i_{k'_x}) + \frac{N_k-1}{N_k}(\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x}))^2] \tag{4.40}$$

$$= 4N_k(\frac{N_k-1}{N_k^2}2\sigma_m^2 + \frac{(N_k-1)^2}{N_k^2}2\sigma_m^2) \tag{4.41}$$

$$= 8\sigma_m^2(\frac{(N_k-1)^2 + N_k - 1}{N_k}) = 8\sigma_m^2(\frac{(N_k^2 - 2N_k + 1) + N_k - 1}{N_k}) \tag{4.42}$$

$$= 8\sigma_m^2(\frac{N_k^2 - N_k}{N_k}) = 8(N_k - 1)\sigma_m^2 \tag{4.43}$$

Similarly, the expected values of the second and third diagonal entries are given by

$$\mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_y})^2] = \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_z})^2] = 8(N_k - 1)\sigma_m^2. \tag{4.44}$$

To compute the expected value of the off diagonal entries, consider the kth product in

the sum in the approximated Hessian (4.36):

$$\mathbb{E}[(2d_{k_x})(2d_{k_y})] = 4\mathbb{E}[(d_{k_x})(d_{k_y})] \tag{4.45}$$

$$= 4\mathbb{E}[((t_x - \overset{*}{t}_x) + (\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x}))((t_y - \overset{*}{t}_y) + (\mathbf{v}^i_{k_y} - \mathbf{v}^j_{k_y}))] \tag{4.46}$$

$$= 4(\mathbb{E}[(t_x - \overset{*}{t}_x)(t_y - \overset{*}{t}_y)] + \mathbb{E}[(t_x - \overset{*}{t}_x)(\mathbf{v}^i_{k_y} - \mathbf{v}^j_{k_y})] + \tag{4.47}$$

$$\mathbb{E}[(\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x})(t_y - \overset{*}{t}_y)] + \mathbb{E}[(\mathbf{v}^i_{k_x} - \mathbf{v}^j_{k_x})(\mathbf{v}^i_{k_y} - \mathbf{v}^j_{k_y})]).$$

Using (4.23), and noting that $\mathbb{E}[\mathbf{v}^i_{k_x}, \mathbf{v}^j_{k_y}] = 0$ for all i and j because the measurement noise values are assumed to be independent, the expected value of each product and the off diagonal sums are zero.

Given the the approximated Hessian is a symmetric matrix, similarly the expected values for the rest of off diagonal entries are given by

$$\mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_x})(2d_{k_z})] = \mathbb{E}[\sum_{k=1}^{N_k}(2d_{k_y})(2d_{k_z})] = 0 \tag{4.48}$$

Thus, the expected value of the approximated Hessian is

$$\mathbb{E}[H] = \begin{bmatrix} 8(N_k - 1)\sigma_m^2 & 0 & 0 \\ 0 & 8(N_k - 1)\sigma_m^2 & 0 \\ 0 & 0 & 8(N_k - 1)\sigma_m^2 \end{bmatrix}. \tag{4.49}$$

The analytical results show that regardless of the geometry of the environment, when the point clouds are perfectly matched (having perfect correspondences), the optimal translation $\mathbf{t}$ is unbiased, and as the number of corresponding points N_k increases, the value of the Hessian approaches a scaled identity matrix with a condition number of $\kappa = 1$. However, for a small number of points, due to random variability of the specific set of samples, variations in the values of the diagonal entries are expected. This will cause the condition number to exceed 1 even when correspondences are perfect. The

following observations can be derived from the analytical study:

- The analytical results in (4.21) and (4.49) show that in case of perfect point correspondences, regardless of the geometry of the environment, the optimal translation $\mathbf{t}$ is unbiased, and for a sufficiently large number of corresponding points the approximated Hessian approaches a scaled identity matrix.

- Since the Hessian H in (4.49) is a diagonal matrix, the eigen decomposition of H leads to three equal valued eigenvalues $\lambda_1 = \lambda_2 = \lambda_3 = 8(N_k - 1)\sigma_m^2$, and a randomly oriented set of eigenvectors. Thus, in this scenario, regardless of geometry of the scene, the condition number is $\kappa \simeq 1$.

- While in the presented analysis of Hessian eigenstructure (4.49) the robot motion was assumed to consist of pure translation, in real-world experiments, the motion usually consists of translation and rotation. However, given that the lidar operates at a high temporal sampling rate relative to robot motion, the relative rotation between two consecutive lidar scans is expected to be very small such that $cos(\theta) \simeq 1$ and $sin(\theta) \simeq \theta$.

- Only when the number of corresponding points is small, due to random variability of the specific set of samples, the condition number will be very large. Thus, for a small number of samples there could be a false indication that the environment is degenerate even if the environment is observable.

In a realistic scenario, the correspondences between two point clouds are never perfect. The probability of mismatching points by the ICP algorithm is further exacerbated by the geometric degeneracy of the scene (e.g., tunnel or corridor with flat and symmetrical walls), where matching ambiguity is increased due to high local point density or variability in the overlapping parts. In this case, the environment becomes unobservable

along the direction of tunnel and this unobservability is expected to manifest itself in the eigenstructure of the Hessian H. The condition number of H is defined as the ratio of the largest eigenvalue λ_{max} to the smallest eigenvalue λ_{min} as given by $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$. Extensive simulation analysis presented in Section 4.2 shows that the large value of the condition number corresponds to the ill-conditioning of the Hessian matrix, and, therefore can be used as a reliable metric to determine geometric degeneracy in an unknown environment.

4.2 Simulation Analysis of Geometric Degeneracy

Geometric degeneracy can be described as the lack or sparsity of distinctive geometric structures in a 3D environment that leads to unobservability of the robot motion along particular directions. In this environment, lidar scan matching using the ICP algorithm can lead to poor alignment of the point clouds due to ambiguity of the environment. This ambiguity leads to an increased number of mismatched point pairs which subsequently leads to an inaccurate estimation of 3D motion between the point clouds.

We can formally define a geometrically degenerate scene as an environment where for any observed 3D landmark $\mathbf{x}^i = [x\ y\ z]^T$, there is a high probability that any landmark $\mathbf{x}^j = \mathbf{x}^i + \mathbf{t}$ is also observed. The optimal computed translation obtained from ICP lidar scan matching can be formulated as

$$\mathbf{t} = \psi\upsilon + \upsilon' \odot \overset{*}{\mathbf{t}}, \tag{4.50}$$

where $\upsilon \in \mathbb{R}^3$ is a binary vector that corresponds to direction(s) of degeneracy in the environment, and υ' denotes the 1's complement of υ, and $\overset{*}{\mathbf{t}} = [\overset{*}{t}_x\ \overset{*}{t}_y\ \overset{*}{t}_z]$ denotes robot's motion consisting of only a translation. An entry with value of 1 in υ indicates geometric

degeneracy in the corresponding direction in the environment. The symbol $\odot$ denotes the element-wise multiplication of the vectors, and ψ is a random scalar, drawn from a two sided truncation of a Gaussian distribution with zero mean and variance σ_ψ^2, where the lower and upper bounds of the Gaussian distribution are based on the field of view of the lidar scanner.

We consider a simple degenerate scene, where a robot moves on a flat surface where there is only a long, and featureless wall present on the left side of the robot. Given a world coordinate system where the x axis points forward, the y axis points to left and the z axis points upward, the robot moves along the x axis parallel to the wall while obtaining lidar scans using the onboard 3D lidar scanner. Considering two lidar scans P^i and Q^j obtained at robot poses i and j, without loss of generality it can be assumed the x and z components of the 3D points are uniformly distributed, while all points are coplanar such that the y component of their position is given by $p^i_{k_y} = \delta_y + v^i_{k_y}$, where $p^i_{k_y}$ is the y component of the kth point in point cloud P^i, δ_y is a constant that corresponds to the relative distance of the wall from the lidar sensor, and $v^i_{k_y}$ is the additive measurement noise. Using the general definition of degeneracy in (4.50), geometric degeneracy along the x axis can be formally defined as given by

$$\mathbf{t} = \psi \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \odot \begin{bmatrix} \overset{*}{t}_x \\ \overset{*}{t}_y \\ \overset{*}{t}_z \end{bmatrix} = \begin{bmatrix} \psi \\ \overset{*}{t}_y \\ \overset{*}{t}_z \end{bmatrix}, \tag{4.51}$$

such that $\psi < \overset{*}{t}_x$. As discussed in section 3.3, downsampling the lidar point cloud is a crucial step to increase the performance and accuracy of lidar odometry. For a low density of points the probability of mismatching between the points is smaller, whereas, for a high density of points the probability of mismatching is higher, as it is more likely that incorrect matched points are closest.

In an environment with higher level of geometric observability (i.e., a feature-rich indoor environment with salient geometric structures) points in an obtained point cloud are not constrained to a particular planar surface, and are more uniformly distributed in the 3D environment. This uniform distribution of points in the 3D space results in a low *volume density* of points on any particular planar surfaces which in turn reduces the probability of point pair mismatching when performing lidar scan registration using the ICP algorithm. In contrast, a *constrained scene* with high level of geometric degeneracy can be considered as an environment with reduced dimensionality such that all the lidar points are constrained to a particular planar surface (e.g., a flat and featureless wall) leading to a higher *area density* of points. The sparsity of salient geometric features leads to higher density of points and a higher probability of point pair mismatching which subsequently results in a poor estimation of 3D motion between the lidar scans. In other words, in a geometrically degenerate scene due to higher local density of points, it is more likely to find a coplanar and colinear corresponding point $p^i_{k'}$ for a reference point q^j_k, such that the Euclidean distance between $p^i_{k'}$ and q^j_k is smaller than the Euclidean distance between the true correspondences p^i_k and q^j_k. In the next section, the impact of geometric degeneracy on the eigenstructure of the Hessian (4.57) and accuracy of estimated 3D motion is studied using extensive simulations of environments with different levels of geometric degeneracy.

4.2.1 Simulation Setup

In this section, through a series of MATLAB simulations the impact of geometric degeneracy on the accuracy of odometric estimates and eigenstructure of the Hessian of the lidar scan matching cost function defined in (4.57) is studied for several environments with a variety of geometric degeneracy levels and 3D dimensions typical to a tunnel and an indoor hallway. Table. 4.1 and Table 4.2 provide the list of all simulated environ-

Table 4.1: List of simulated featre-rich Tunnel environments.

Name of Tunnel	X Dimension (m)	Y Dimension (m)	Z Dimension (m)
T1	80	10	10
T2	40	10	10
T3	20	10	10
T4	10	10	10
T5	40	20	10
T6	20	10	10
T7	20	20	10

Table 4.2: List of simulated featureless Wall environments.

Name of Wall	X Dimension (m)	Y Dimension (m)	Z Dimension (m)
W1	40	0	10
W2	40	1	10
W3	40	5	10
W4	10	0	10
W5	10	1	10
W6	10	5	10

ments. The tables show two types of simulated environments, namely a feature-rich tunnel environment with high geometric observability where lidar points are uniformly distributed in the 3D space, and a feature-poor and geometrically degenerate flat wall environment, where points are uniformly distributed on the wall so the environment is reduced to two dimensions as the only reflector in the environment is a flat wall. For each environment type, a variety of 3D dimensions are considered while the number of simulated lidar points is kept constant at $10,000$ points. This enables the study of impact of point cloud density on the eigenstructure of the Hessian as well as the accuracy of the point cloud aliments. For instance, $T1$ is a feature-rich tunnel where $10,000$ points are distributed uniformly in a tunnel that is 80m long, $10m$ wide, and 10 m tall, whereas, in $W1$ all the points are distributed uniformly on a 2D wall that has a length of 40m and a height of 10m. Where the Y dimension is given a non-zero value, the simulated flat wall becomes slightly more feature-rich where points are reflected off the

wall from different depths. Fig. 4.2, presents examples of simulated environments where point clouds P and Q each with $10,000$ points are highlighted in green and red.

In each experiment, two set of point clouds, P and Q, are obtained by sub-sampling the simulated environments. In order to simulate the lidar measurement noise, uncorrelated Gaussian noises $\mathbf{v}^i, \mathbf{v}^j \sim \mathcal{N}(0, \sigma_m^2)$ are added to all the points in P^i and Q^j using MATLAB's *rand* function such that

$$p_k = \mathbf{p}_k + \mathbf{v}_k^i, \quad \text{for all} \;\; p_k \in P^i \tag{4.52}$$

$$q_k = \mathbf{q}_k + \mathbf{v}_k^j, \quad \text{for all} \;\; q_k \in Q^j \tag{4.53}$$

Subsequently, a transformation $\mathbf{T}$ consisting of only a forward translation along the x axis with a constant value of $t_x = 0.1$m is applied to all the points in Q to simulate the forward motion of the robot in the environment between consecutive key-scans. Larger translation values of 0.2m, 0.4m, 0.6m, and 0.8m were also tested which lead to similar results, while translation values of $t_x \leq 1.0$ m, increased the alignment error

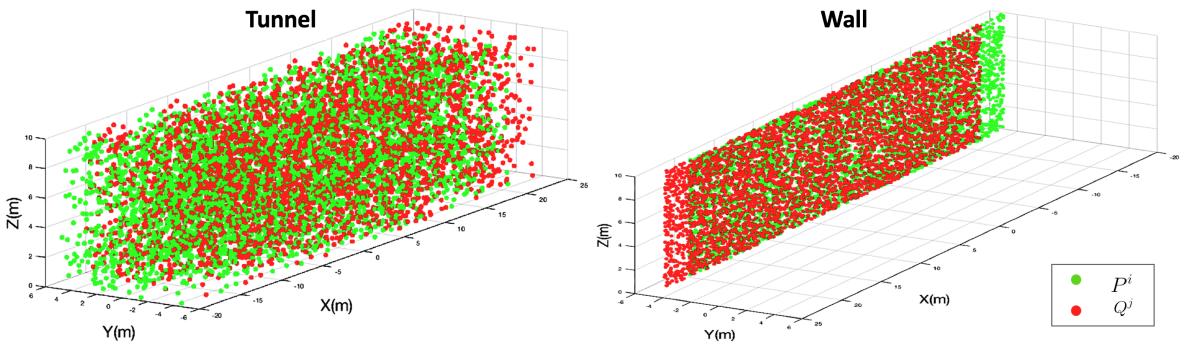

Fig. 4.2: MATLAB simulation of observable and degenerate scenes. (a) A feature-rich tunnel environment: simulation of an observable environment, where $10,000$ points are uniformly distributed in the 3D space. (b) A geometrically-degenerate wall environment: simulation of an environment with geometric degeneracy where $10,000$ points are constrained to a planar surface and uniformly distributed across the x and z axis.

and uncertainty in the estimated motion between the point clouds due to poor initial alignment of point clouds which made the ICP algorithm susceptible to local minima.

MATLAB's point cloud registration tool based on the ICP algorithm [97] is used to align point sets and to compute the optimal transformation between them. Once two point clouds are aligned, the algorithm returns the following information:

- a rigid transformation $\mathbf{T}$ that best aligns the two point clouds P and Q
- the transformed point cloud Q' that is aligned with the reference point cloud P,
- and the mean squared error (MSE) of the Euclidean distance between the aligned point clouds.

Finally, using the estimated set of corresponding points and the rigid transformation matrix $\mathbf{T}$ returned by the ICP algorithm, the Hessian H (4.57) of the cost function (4.11) is computed from the sum of outer product of the gradient vectors J from (4.12) between all pairs of corresponding points. The condition number κ of the Hessian is then obtained from the eigendecomposition of the Hessian to analyze the level of geometric degeneracy in the environment. All experiments are repeated 50 times and the results show the mean value for each metric.

The three key metrics evaluated in the simulations are:

- Condition number: the condition number of the approximated Hessian H
- Transform error percentage: the error percentage between the optimal 3D transformation between two point clouds obtained from the ICP algorithm and the ground truth
- Mean squared error (MSE): the residual error between the aligned corresponding points after point cloud registration determined by the ICP algorithm

4.3 Analysis of the ICP Error Metric

In this simulation, the ICP error metric (i.e., the fitness score) is analyzed in environments with different levels of geometric degeneracy. This is the cost function that is iteratively minimized by the ICP algorithm as it attempts to find the best alignment between two lidar scans. As presented in Fig. 4.3, five environments with different levels of geometric degeneracy are simulated by combining points from the simulated tunnel and the wall environment. For instance, the 0% degeneracy indicates that all the points are sampled from the simulated feature-rich tunnel environment, whereas, the 100% degeneracy indicates that all the points are sampled from the featureless wall environment. The results show, in environments with higher level of geometric degeneracy, the ICP error metric which reports the mean squared error between aligned point clouds is significantly smaller as compared to environments with lower level of geometric degeneracy. This is due to the fact that as the size of the point cloud increases, the probability of mismatches in corresponding points also increases, however, in the featureless wall environment where all the points are concentrated on a flat wall, increasing the point density or the measurement noise also means the Euclidean distance between the points is further reduced. So even though the probability of mismatches is significantly higher in the wall environment, the mean squared error between the mismatched points

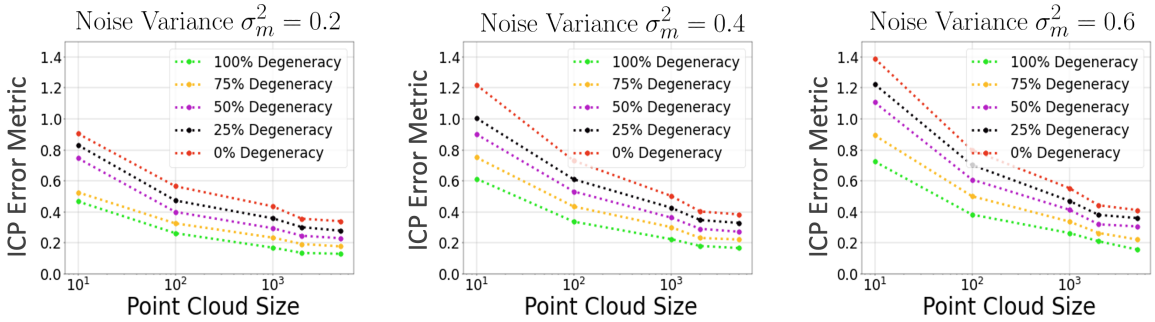

Fig. 4.3: Impact of variations in levels of geometric degeneracy, point cloud size and measurement noise on the ICP error metric.

is smaller due to larger number of points in the same volume which further reduces the distance between the points.

In the simulated feature-rich tunnel where the points are uniformly distributed in 3D space, the point density per volume is lower. In this environment, increasing the size of the point cloud leads to higher probability of mismatches which in turns leads to higher error values as the distance between every mismatched point pair is larger as compared to the wall environment. Another conclusion that can be derived is that as the ICP algorithm has no knowledge of the actual motion between the point clouds, it only relies on the MSE between the aligned points to assess the convergence to the optimal solution. However, as the results show *a low ICP error does not necessarily indicate a low actual error* in alignment of the points, thus, in a geometrically degenerate scene where the ICP algorithm is more susceptible local minima this could lead to inaccurate estimation of motion between lidar scans.

Fig. 4.4 presents the percentage error Δt between the computed and actual rigid transformation between point clouds obtained by the ICP algorithm as given by

$$\Delta t = \frac{\overset{*}{t} - t}{\overset{*}{t}} \times 100\%. \tag{4.54}$$

The results show in environments with higher level of observability where points are uni-

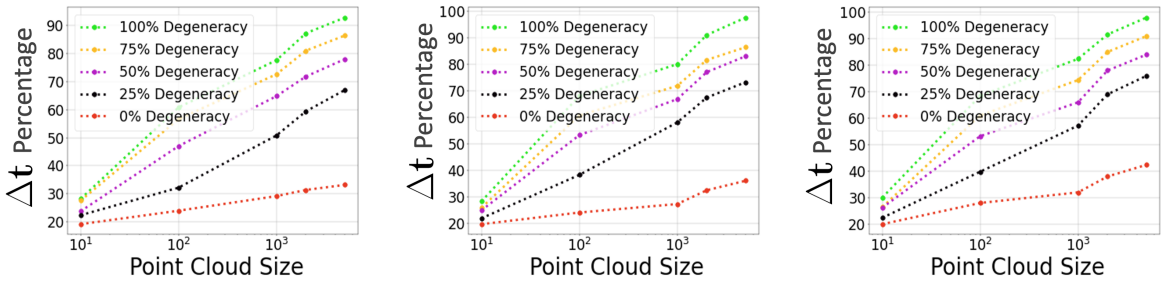

Fig. 4.4: Impact of variations in environment degeneracy, point cloud density and lidar measurement noise on the ICP error metric.

formly distributed in 3D space the probability of mismatches is significantly smaller as compared to geometrically degenerate scenes. Thus, the estimated rigid transformation between lidar scans is more accurate yielding to a smaller transform error percentage. The result further highlights the fact that the mean squared error reported by the ICP algorithm cannot be used as a reliable error metric in evaluating the reliability of lidar scan matching and determining the level of geometric degeneracy in unknown environments. This underscores the importance of developing an error metric that responds to the actual error and unreliability in the ICP estimations as the result of high levels of geometric degeneracy in the environment.

Fig. 4.5, reports the log of the condition number as a function of point density and a measurement noise variance of 0.6 that is a representative noise value for the two simulated environments with different 3D dimensions. The results confirms the symbolic analysis of the Hessian (4.49); when the number of corresponding points is too small (i.e., 10 points in each point cloud), due to random variability of the specific set of samples, the condition number tends to be larger. Thus, for a small number of samples there could be a false indication that the environment is degenerate as it is more likely to get different behaviors in different directions when the number of points is small. As the number of points increases, variability in Hessian elements is reduced so the eigenvalues tend to more similar values and condition number decreases. However, as

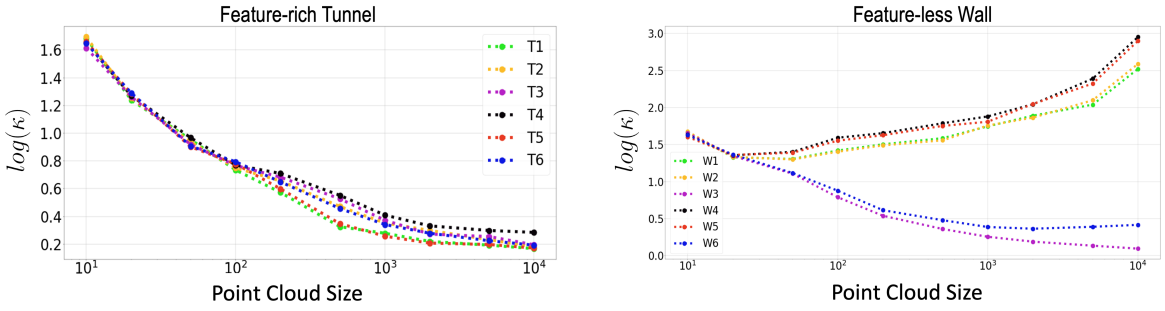

Fig. 4.5: Impact of density of points on the condition number κ, in feature-rich tunnel and featureless wall environments with varying dimensions of the physical environment.

number of points increases further in the featureless wall environment, the probability of mismatches becomes higher due to differential density of points in the wall models, so the mismatches occur preferentially in the high density dimension. In the wall environment, when the Y dimension of the volume is set to zero or is close to zero, the simulated scene is a flat and featureless wall. In W3 and W6 the Y dimension of the wall is increased to simulate a scene with some geometric variations on the wall. The result show when the wall is completely flat or close to flat, the proposed metric κ is a very strong indicator that the ICP algorithm is returning plausible matches, which are *not* correct. The reason for this increase in the condition number is that, the high density of points give directional concentration of mismatches along the wall, thus, the structure of Hessian is no longer a scaled diagonal matrix as we saw in the mathematical analysis. In the simulated feature-rich tunnel environment, where the points are uniformly and sparsely distributed in the environment, the probability of finding correct correspondences by the ICP algorithm is increased, thus, the Hessian is close to a scaled identity matrix for an increasing number of correspondences leading to smaller values of κ.

Fig. 4.6 shows the log of the condition number as the result of variations in levels of geometric degeneracy in the scene, point density, and lidar measurement noise variances. The results show that κ reliably responds to unconstrained situations where due to geometric degeneracy of the scene and measurement noise the number of false correspondences between two point clouds increases as the result of increased randomness in finding the true correspondences. With increased levels of geometric degeneracy (i.e., larger density of points constrained to a planar surface), $log(\kappa)$ monotonically increases in all simulated degenerate environments. In environments with higher observability points have a lower density as they are uniformly distributed over a large volume, thus, increasing the number of points does not lead to substantial increase in point density and noticeable changes in values of κ. Moreover, it can be observed that increasing

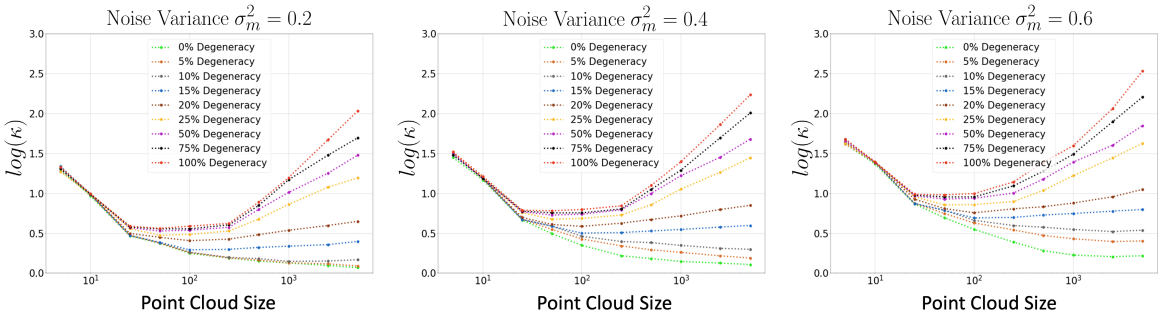

Fig. 4.6: Response of the $log(\kappa)$ to variations in geometric degeneracy, point cloud density and lidar measurement noise. (a): $\sigma_m^2 = 0.2$, (b): $\sigma_m^2 = 0.4$, (c): $\sigma_m^2 = 0.6$

the lidar measurement noise in all the simulated environments leads to more incorrect correspondences which subsequently increases κ.

The following conclusions can be derived from the simulation analysis:

- The alignment error provided by ICP is not a reliable metric to evaluate the confidence and reliability of ICP solution.
- A low ICP error does not indicate a low actual error in the computed optimal transformation.
- Condition number of the Hessian of error cost function can be used to differentiate areas with high and poor geometric observability in an unknown environment by evaluating the confidence level in the solution of ICP algorithm.
- From mathematical analysis the ideal value for the condition number in case of perfect correspondences is determined which enables setting decision thresholds for determination of geometric degeneracy in unknown environments.

4.4 Experimental Results and Analysis

In this section, the response of the condition number to geometric degeneracy is evaluated using data collected in an indoor office environment, as well as an underground mine. From the scan-to-submap matching step of the lidar odometry presented in Section 3.3.2, a rigid transformation is obtained that describes the 3D motion between two consecutive lidar scans. Using the displacement vector from (3.1), the ICP cost function is be defined in terms of $\mathbf{d}_k$, the displacement error vector computed for the kth pair of corresponding points at time step $t+1$. Analogous to (4.11), the squared displacement vector is

$$\xi_k(\mathbf{u}_{t+1}^t) = d_{k_x}^2 + d_{k_y}^2 + d_{k_z}^2. \tag{4.55}$$

Assuming local linearity, the gradient of the cost function is computed to propagate errors from sensor noise to uncertainty of state estimates. In the previous development, (4.12) only included translation. In a more general case where the rigid transformation includes rotation, the gradient vector of the cost function is given by

$$J_k^{t+1} = \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial(t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)} = \begin{bmatrix} \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial t_x} & \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial t_y} & \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial t_z} & \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial \theta_x} & \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial \theta_y} & \frac{\partial \xi_k(\mathbf{u}_{t+1}^t)}{\partial \theta_z} \end{bmatrix}. \tag{4.56}$$

Here J_k^{t+1} is the gradient of the scalar-valued cost function (4.55). The approximated Hessian [96] for the set of N_k corresponding points is a 6×6 matrix that is obtained from the sum of outer products of the gradient vectors as given by

$$H^{t+1} = \sum_{k=1}^{N_k} (J_k^{t+1})^T \, J_k^{t+1}. \tag{4.57}$$

Note that as shown in Section 4.5, H is not the true Hessian (second derivative) of the ICP cost function, and is approximated by neglecting second (and higher) order derivatives. In order to obtain the numerical representation of environment degeneracy κ, the eigendecomposition of the Hessian matrix is obtained. In the eigenvalue decomposition of H^{t+1}, the eigenvector associated with the smallest eigenvalue represents the least observable direction in terms of the estimated rigid transformation from the ICP algorithm. We find the relative scale between the most and least observable directions by computing the ratio of the maximal and minimal eigenvalues as given by

$$log(\kappa^{t+1}) = \frac{|\lambda_{max}|}{|\lambda_{min}|}. \tag{4.58}$$

Although the eigenvalues will be dependent on general environmental variables such as measurement noise or the number of corresponding points in the obtained lidar scans, computing the ratio will remove dependence on environmental variables that scale the eigenvalues. Thus, a large condition number corresponds to higher levels of geometric degeneracy [98]. This means a relatively large error is associated with one or more linear combination of parameters of the computed transformation $\mathbf{u}_{t+1}^{t}$. A small condition number close to 1 corresponds to equal observability of all parameters of the computed transformation.

4.4.1 Degeneracy analysis in an indoor environment

In this experiment, the geometric degeneracy is determined by analyzing the eigenstructure of the Hessian H in an indoor office environment. A Husky A200 ground rover equipped with a 3D lidar scanner is used to obtain lidar scans at 10 Hz as the robot traverses a hallway with flat, symmetric and featureless walls. In these experiments, a scene is classified as degenerate if $\log(\kappa^{t+1}) \geq \kappa_{th}$, where κ_{th} is the degeneracy

threshold based on expectations of normal variability and validated by experimental results. While an indoor office can be considered as a feature-rich environment in some respects, the symmetric and featureless corridors create a challenging geometric scene for the lidar-based front-end, where all 3D points are constrained to symmetric planar surfaces, leading to a situation we call the *lidar-slip*.

Fig. 4.7 shows the detection of geometric degeneracy in the office environment as a robot navigates a long corridor with flat and symmetric walls. The plot of log of the eigenvalues $log(\lambda)$ shows the response of all six eigenvalues of the approximated Hessian matrix. As the robot navigates the featureless corridor, a significant drop can be seen in the values of the smallest eigenvalue of the Hessian, while variations in the rest of eigenvalues are minimal. In one section of the trajectory between points B and C, this leads to a significant rise in values of the condition number in (4.58). In this experiment, it is possible to verify that increased values of κ indeed corresponds to

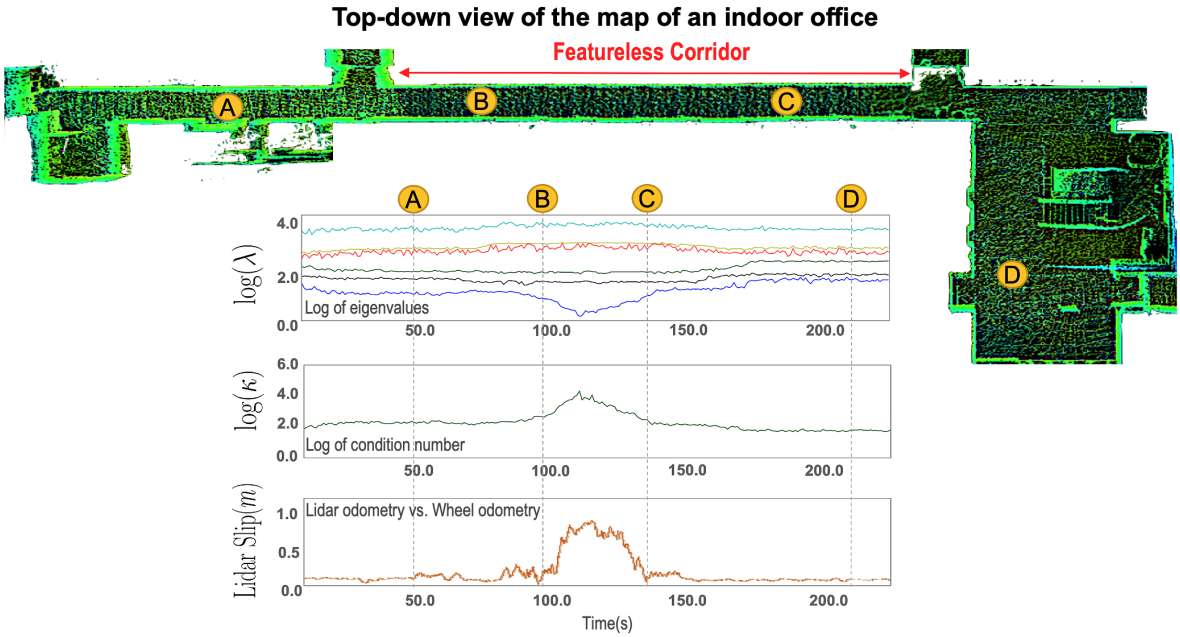

Fig. 4.7: Impact of geometric degeneracy on eigenvalues of the Hessian, condition number, and lidar slip. The 3D map shows the top-down view map of the indoor office environment.

inaccurate odometric estimates based on the wheel-inertial odometry. We compare the output of the lidar odometry with the wheel-inertial odometry as the robot navigates the indoor office environment. Often the wheel odometry is not reliable due to wheel slippage, but in this case as the robot moves at a slow speed and office has carpet flooring the wheel slippage is negligible. The lidar slip plot shows the difference in the estimated forward motion as reported by the wheel-inertial odometry and the lidar odometry at 1s intervals. The results show values of $\log(\kappa) > 2.0$ correspond to the largest discrepancies between lidar and wheel-inertial odometry.

Fig. 4.8 reports the plots of all six eigenvalues and their histograms. The result shows as the robot navigates the corridor, there is a significant decrease in the values of the smallest eigenvalue, λ_6, while the rest of eigenvalues show minimal variations and λ_1 is at the high of its range. This is due to the fact that in this scene, the geometry of the environment does not fully constrain the 6-DOF estimated motion of the robot. This means while the observability along different dimensions stays relatively stable, the

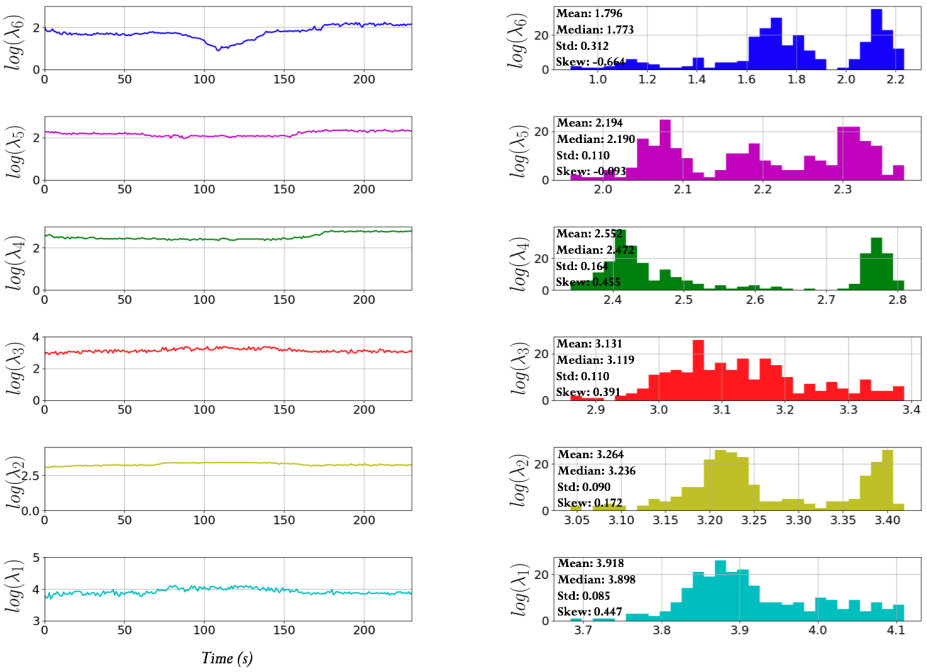

Fig. 4.8: Plots and histograms of eigenvalues of the Hessian matrix.

observability of robot motion parallel to the walls of the corridors varies significantly depending on the geometrical structure of the environment. This is evident in the eigenvalue analysis presented in Fig. 4.8 where other than the smallest eigenvalue which corresponds to the dimension with lowest observability, the rest of eigenvalues show minimal variations as the robot navigates the environment.

The scatter plots in Fig. 4.9 show the correlation between the smallest eigenvalue and other eigenvalues of the Hessian. The results show the condition number κ, can reliably detect the degeneracy in the environment as the largest eigenvalue λ_1 has minimal variations throughout the course of travel, while λ_6 decreases dramatically when the level of geometrically degeneracy is increased.

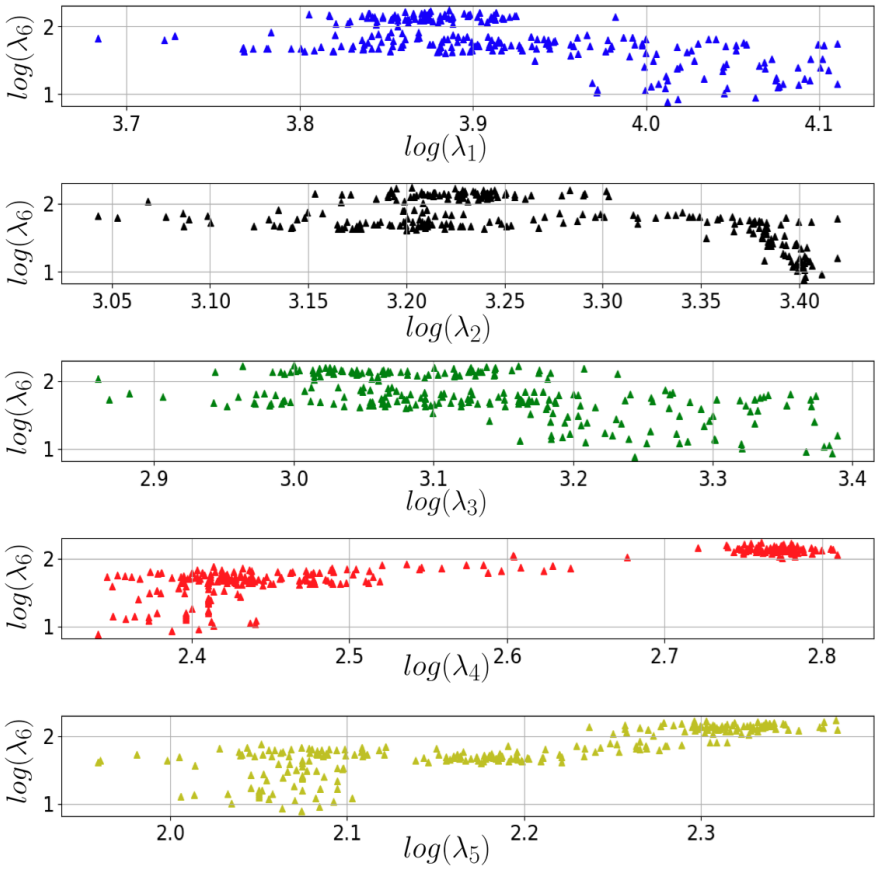

Fig. 4.9: 2D scatter plots of eigenvalues of the Hessian matrix.

Given that in real-world conditions under real physical constraints, angular variations between consecutive lidar scans obtained at high sampling rates are small, the 3×3 upper left block of the Hessian matrix that corresponds to the translation component of the estimated rigid transformation can be analyzed to determine the response of the eigenvalues to geometric degeneracy in the environment. Fig. 4.10 which reports the plots and histograms of the three eigenvalues shows a significant drop in the values of λ_3 which corresponds to the least observable dimension in the estimated translation vector, while λ_1 and λ_2 show minimal variations indicating consistent observability of the environment along the other dimensions. The log of the condition number which in this case is given by $log(\kappa) = \lambda_1/\lambda_3$, shows the response of the condition number to the translational observability in the environment as the robot navigates the office space. Similar to the eigen analysis of the Hessian matrix the results show, loss of observability of robot's forward motion along the walls can be reliably detected in the eigen analysis of the translation block of the Hessian matrix. The result does not show any improvement by including the rotation component of the Hessian in the analysis, thus, the analysis of geometric degeneracy can be limited to eigendecomposition of the

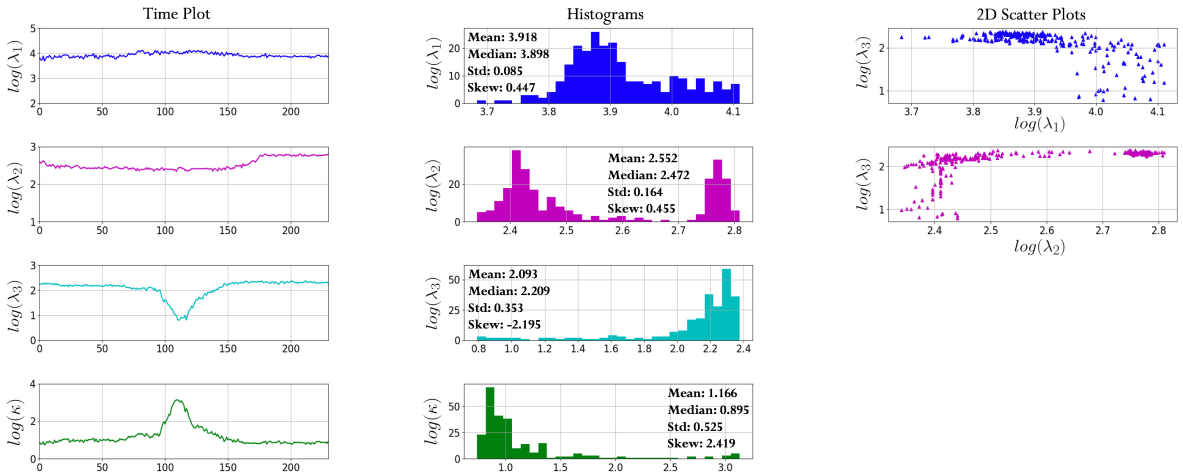

Fig. 4.10: Eigenvalue analysis of the 3×3 upper left block of Hessian corresponding to the translation component of the estimated 3D transformation between two point clouds.

translation component of the Hessian.

Fig. 4.11 is a visualization of the eigenvectors and eigenvalues of the 3×3 block of Hessian corresponding to translation component of the estimated 3D transformation between two point clouds, where red arrows $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ are eigenvectors and their magnitude correspond to the log of the corresponding eigenvalue. Fig. 4.11(a)-(b), show the side and top view of the 3D map of the corridor in the indoor office environment. The results show the presence of the geometric features in the tunnel has contributed to the observability of the motion along the tunnel axis. In Fig. 4.11(c), where most of the environment is flat and featureless the observability along the corridor is significantly reduced resulting in a drop in the values of λ_3. This can be seen in vector $\mathbf{v}_3$ with a significantly smaller magnitude as compared to Fig. 4.11(a)-(b). By observing the response of eigenvalues to the variations in the levels of geometric degeneracy in the environment, in addition to using the condition number, two other metrics can be considered for detection of geometric degeneracy. Given that the determinant of H is

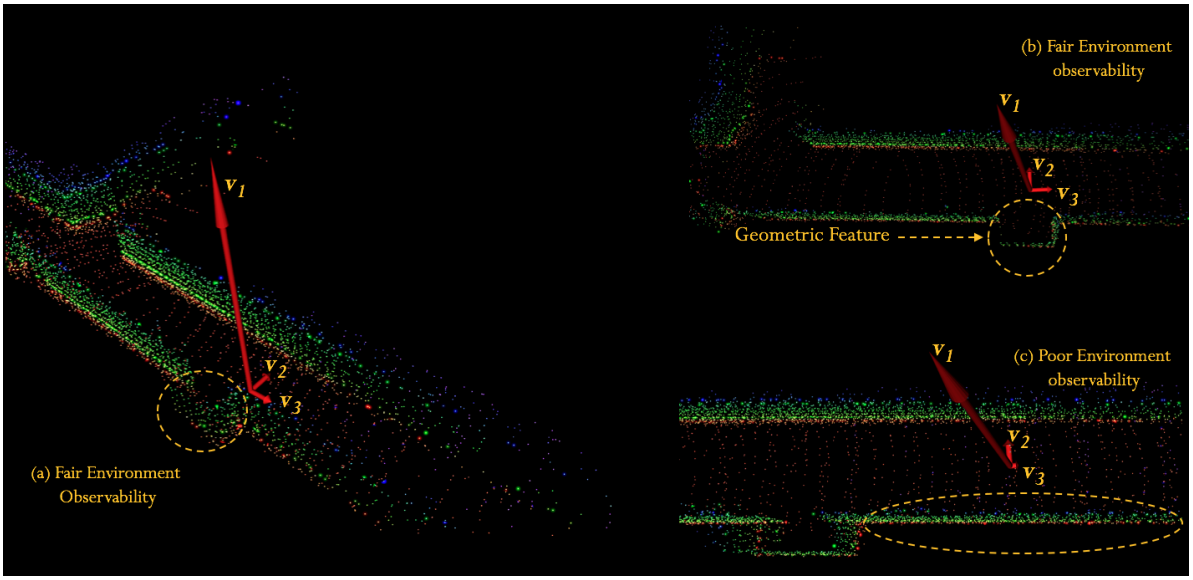

Fig. 4.11: Visualization of the eigenvectors and eigenvalues of the 3×3 block of Hessian corresponding to translation component of the estimated 3D transformation between two point clouds.

equal to product of its eigenvalues, both the determinant of H and determinant of the 3×3 translational block of H can be considered as metrics to determine variations in the level of environment degeneracy where a binary classifier can be used to classify a scene as degenerate when the value of a metric exceeds a set threshold. Fig. 4.12 presents the plots for product of all six eigenvalues of H as well as product of the three eigenvalues corresponding to the translational block of the Hessian. The plots show both metrics respond to loss of observability in the environment.

In order to evaluate the performance of each metric, we rely on the Receiver Operating Characteristic (ROC) analysis. ROC analysis is a standard way of evaluating metrics in binary classifier systems to decide how to set decision thresholds. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. In this experiment a set of lidar scans are manually labeled as degenerate and observable such that there are 61 lidar scans obtained in degenerate scenes (true positive) and 193 lidar scans obtained in areas with high geometric observability (true negative). The three evaluated metrics are named $M1$ for the condition number

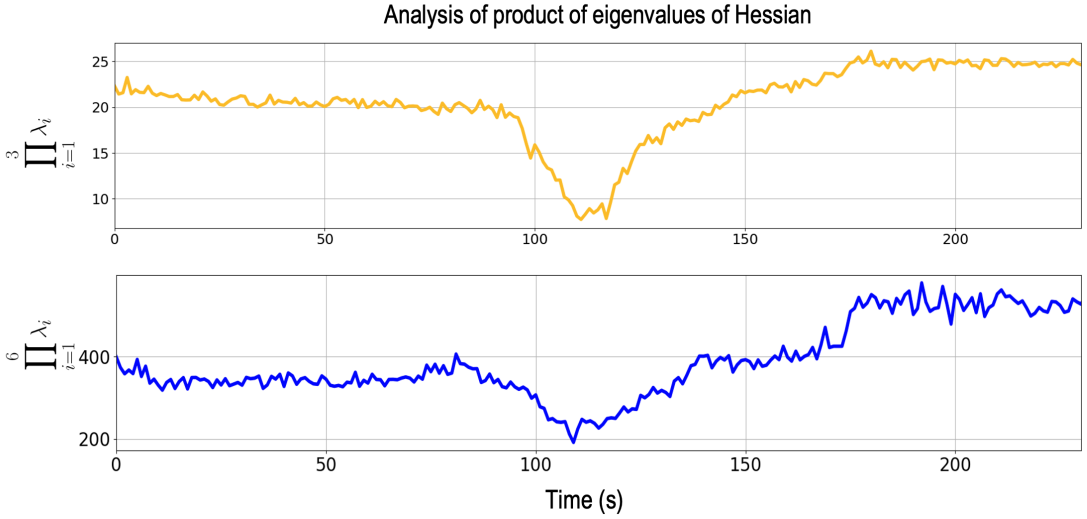

Fig. 4.12: Plots of product of all six eigenvalues of H, as well as product of three eigenvalues corresponding to the translational component of the estimated 3D transformation between two point clouds.

κ of the translation block of Hessian, $M2$ for the product of *log* of three eigenvalues of the translation block of Hessian, and $M3$ for the product of *log* of all six eigenvalues of the Hessian. Using different threshold values, the false positive and true positive rates are determined for each metric.

The ROC plot presented in Fig. 4.13 reports the true positive and false positive rates in classification of environment degeneracy in the indoor office environment. The results show both $M1$ and $M2$ metrics have similar performance with area under curve (AUC) value of 0.887 and 0.862 respectively. As $M3$ is obtained from the product of all 6 eigenvalues of H, small variations in the values of all eigenvalues leads to more variations in the product of eigenvalues, thus, it leads to clearly less accurate detections of geometric degeneracy and subsequently leads to lower AUC values. One advantage of using the product of the three eigenvalues over the condition number is that it can be obtained by computing the determinant of the Hessian matrix, thus, removing the

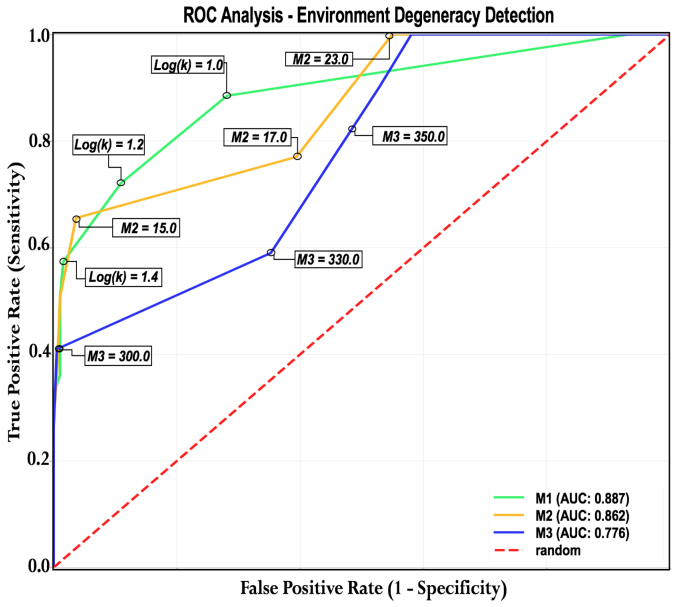

Fig. 4.13: ROC curve reports the performance of environment degeneracy detector in 20 degenerate scenes using three metrics; $M1$: condition number κ, $M2$: product of 3 eigenvalues of the translation block of Hessian, $M3$: product of 6 eigenvalues of the Hessian.

need to perform eigendecomposition after each lidar scan registration, which can reduce the computational load in resource constrained applications.

4.4.2 Degeneracy analysis in an underground mine

In this experiment, the response of the eigenvalues of the Hessian to geometric degeneracy of the scene is further evaluated in the long and narrow tunnels of the Eagle mine with an average width of 1m., Julian, CA. A handheld lidar scanner is used to scan the environment and obtain point clouds at 10 Hz. Throughout the data collection process, by partially blocking the field of view of the handheld lidar scanner four geometrically degenerate segments were manually introduced at separate locations. Limiting the field of view of the lidar creates geometric degeneracy where there are a large density of points reflected off of the walls of the tunnel that are only ≤ 0.5 m away from the lidar scanner, while no lidar reflections are received from the front and back of the lidar as the tunnel is a straight and long corridor with no reflectors at both ends. The high density of points on the planar surfaces of the side walls leads to geometric degeneracy which subsequently leads to poor estimation of 3D motion using the lidar front-end. The partial map of the Eagle mine along with the locations of degenerate segments are shown in Fig. 4.14 where the peak values of $log(\kappa)$ corresponds to the detection of

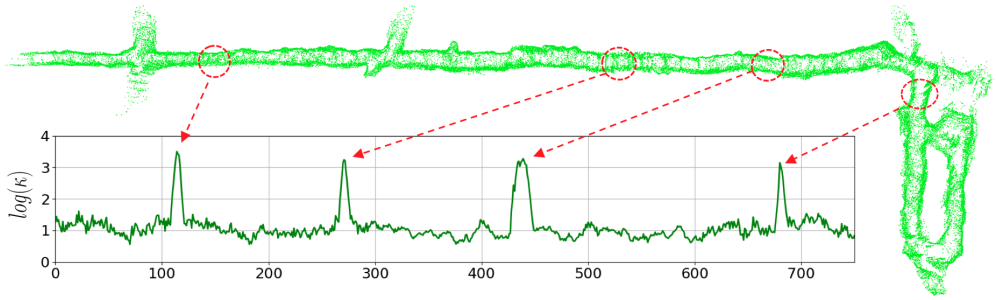

Fig. 4.14: Top: Partial 3D map of the Eagle Mine. Bottom: plot of the response of κ to four degenerate scenes that are manually introduced inside the tunnel.

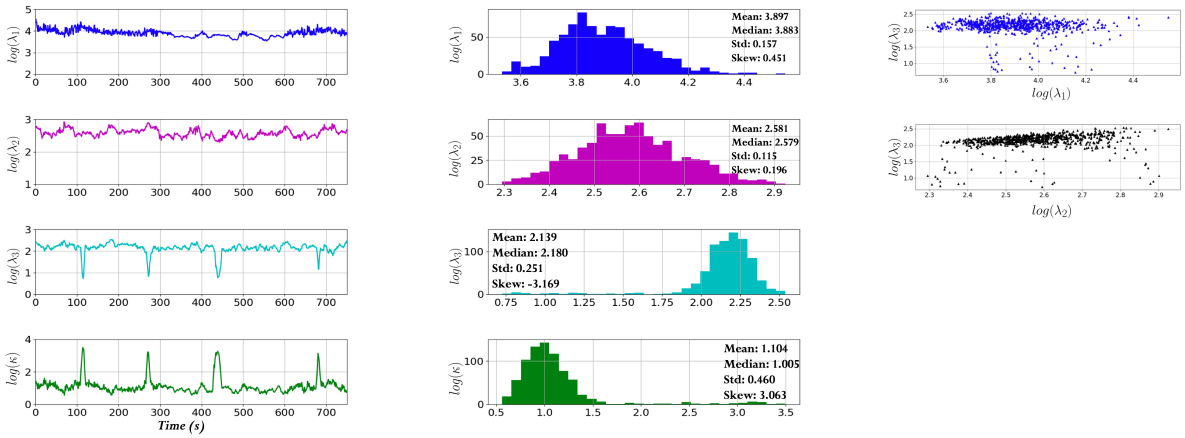

Fig. 4.15: Eagle Mine - Eigenvalue analysis of the Hessian in the Eagle mine. From left to right, time plot, histogram and 2D scatter plots of the three eigenvalues of the translational block of the Hessian matrix.

geometric degeneracy in the mine.

Fig. 4.15 shows the eigenvalues, and condition number κ of the translation block of the Hessian matrix. Similar to the eigenvalue analysis in the indoor environment (Fig. 4.8, Fig. 4.10), the results show the smallest eigenvalue responds to reduced observability along the direction of the tunnel, and it has the largest standard deviation and skew as compared to the other eigenvalues. With a skewness of 3.063, the log of the condition number of Hessian is highly skewed to the right, where values of $log(\kappa) \geq 3.0$ correspond to high levels of geometric degeneracy in the tunnel.

4.5 Approximation of Hessian

Given a nonlinear least squares problem of m functions of $f_i(x)$ of parameters $x = (x_1, \cdots, x_n)$, and $m \geq n$ as given by

$$\underset{x \in \mathbb{R}^n}{\operatorname{argmin}} F(x) = \sum_{i=1}^{m} (f_i(x)^2), \tag{4.59}$$

the gradient of $F(x)$ is $J^T f$, where J is the $m \times n$ Jacobian matrix, whose (i,j)th entry is $J_{ij} = \frac{\partial f_i}{\partial x_j}$, and f is the m-vector function values $(f_1, \cdots, f_m)$. The entries of the Hessian of the cost function are obtained by calculation of the derivative of the gradient as given by

$$H = \sum_{i=1}^{m} (\frac{\partial f_i}{\partial x_i} \frac{\partial f_i}{\partial x_j} + f_i \frac{\partial^2 f_i}{\partial x_i \partial x_j}) \tag{4.60}$$

The first term in the sum is the squared of the Jacobian of f, and the second term is f_i multiplied by the Hessian of f_i as given by

$$H = J^T J + \sum_{i=1}^{m} (f_i \frac{\partial^2 f_i}{\partial x_i \partial x_j}) = J^T J + G \tag{4.61}$$

The evaluation of Hessian is computationally expensive as it requires $\mathcal{O}(N^2)$ function evaluations. In many optimization methods such as the Gauss-Newton and the Levenberg-Marquardt algorithms that aim to minimize a cost function are both based upon the premise that the $J^T J$ term in the Hessian dominates its other terms [99], and that the Hessian may be approximated by the dominant term. Approximation of the Hessian can be obtained with low computational cost as it only requires the computation of the Gradient that requires $\mathcal{O}(N)$ function evaluations. Convergence of the Gauss–Newton method is not guaranteed in all instances. The approximation

$$|f_i \frac{\partial^2 f_i}{\partial x_i \partial x_j}| \ll |\frac{\partial f_i}{\partial x_i} \frac{\partial f_i}{\partial x_j}| \tag{4.62}$$

that needs to hold to be able to ignore the second-order derivative terms may be valid in two cases; (i) the function values f_i are small in magnitude, at least (ii) the functions are only "mildly" nonlinear, $f_i \frac{\partial^2 f_i}{\partial x_i \partial x_j}$ is relatively small in magnitude.

Chapter 5

Loop Closure Detection

In Chapter 3 it was shown how the SLAM front-end can compute the relative motion of a mobile robot by relying on lidar scan registration. In the absence of any prior map of the environment that can be used for global localization, the open loop accumulation of errors from ICP-based lidar scan registration can lead to an unbounded drift in the estimated robot trajectory as error variance increases after each motion step. This drift, is inherent to any open loop odometry system, and it can only be limited and subsequently reduced by detection of loop closures when a robot returns to a previously visited or otherwise known location or landmark. Moreover, in a multi-robot SLAM system, inter-robot loop closures are crucial to find the correspondences between robot trajectories. Each loop closure adds a new nonlinear constraint between the corresponding key-nodes in the pose graphs. With well constrained pose graphs, the robot trajectories can be optimally merged by formulating the problem as the maximum a-posteriori (MAP) likelihood estimation of relative poses in the pose graphs, given the set of all intra- and inter-robot measurements.

In this chapter, a new method is developed to exploit salient semantic and geometric features extracted from lidar point cloud data to improve place recognition and loop

closure detections in single and multi-robot SLAM systems. As presented in Section 3.3.3, a pose graph is constructed based on lidar odometry, where a key-scan is associated with a key-node in the graph. By relying on semantic features extracted from 2D occupancy grid maps that are constructed from the key-scans, the candidate loop closure key-nodes are identified using a pre-matching step. In the next step, the key-scans associated with candidate loop closure key-nodes are used to verify and confirm loop closures using a geometric verification step. Finally, an outlier loop closure detection method is used to reject the outlier loop closures before adding new constraints between the loop closure key-nodes.

5.1 Challenges in Basic Geometric Loop Closures

A common approach in detection of loop closures in a graph-based SLAM system is to register a lidar scan to previously obtained key-scans in order to find a match, where a match is confirmed by evaluating the alignment error between registered point clouds. This basic geometric loop closing (BGLC) method can become increasingly computationally expensive and lead to spurious or inaccurate loop closures in large-scale and ambiguous environments. To remedy this, a common approach is to constrain the search for loop closures to a fixed space centered at the estimated robot poses. As presented in Section 2.3.2 this can be formulated as

$$\left\|\mathbf{x}_{\alpha_i} - \mathbf{x}_{\alpha_j}\right\| < D_r, \tag{5.1}$$

where the positional components of robot poses $\mathbf{x}_{\alpha_i}$ and $\mathbf{x}_{\alpha_j}$ are used to compute the Euclidean distance $\|.\|$ between two robot poses, and D_r is the loop closure search radius. Although this is a reasonable constraint, in practice if the drift in robot trajectory exceeds D_r, loop closure opportunities will be missed. The impact of drift is illustrated

in Fig. 5.1, where the search for loop closure is performed by registering a key-scan to historic key-scans that are associated with the key-nodes that lie inside the loop closure search space. Since the search for loop closures has a fixed radius and is dependent on estimated robot poses, this could lead to missed loop closure opportunities where the drift in robot trajectory is larger than expected.

Fig. 5.1, depicts an example of geometric loop closure detection in local pose graph. The search for loop closures is performed by matching the key-scan associated with

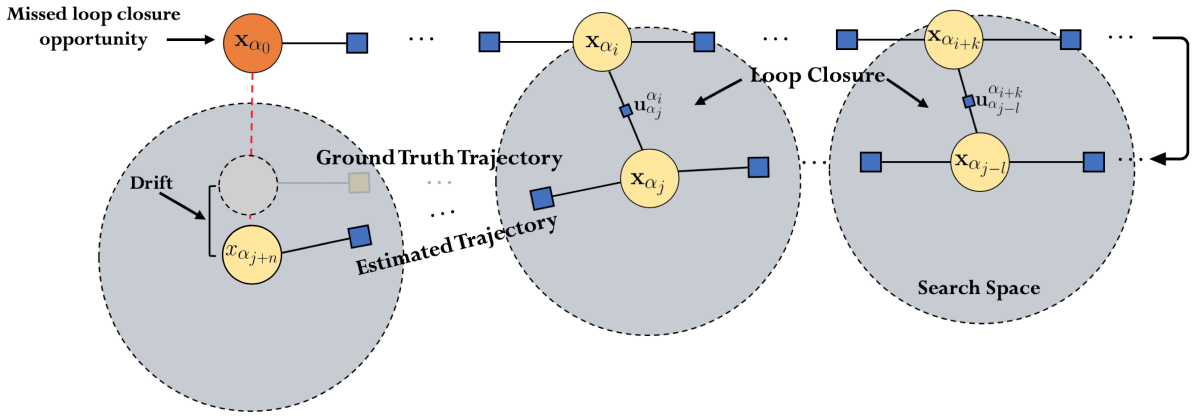

Fig. 5.1: Illustration of basic geometric loop closure detection method.

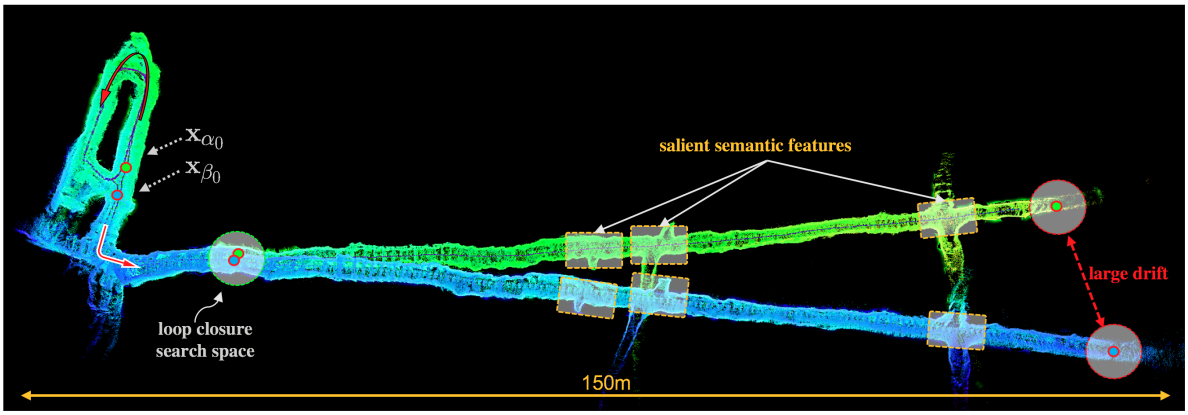

Fig. 5.2: Partial map of the Eagle mine, Julian, CA, obtained on the base station by merging local pose graphs of two unmanned ground robots using the basic geometric loop closing method. Due to significant drift in the estimated trajectories, many loop closure opportunities are missed as shown in unmerged blue and green branches, where both branches represent the same physical environment. The highlighted salient semantic features can be used in a pre-matching step to improve detection of loop closures.

each robot pose with key-scans associated with the historic key-nodes in the graph that lie inside the loop closure search space. The figure illustrates a missed loop closure opportunity due to the significant drift in robot trajectory. While the Euclidean distance between the *true* poses of the robot at node α_{j+n} and $\mathbf{x}_{\alpha_0}$ is less than D_r, but condition (5.1) is not satisfied for the pair $\mathbf{x}_{\alpha_0}$ and $\mathbf{x}_{\alpha_{j+n}}$ due to accumulation of drift in the estimated trajectory.

Fig. 5.2, presents a distorted map of the Eagle mine, where due to drift in robot trajectory two identical branches (shown in green and blue) are created while both correspond to the same physical tunnel in the mine. Due to large drift, many loop closure opportunities are missed leading to the failure of the front-end in merging the two branches. As the environment is unknown and no prior information about the level of perceptual degradation or frequency of physical loop closures is available, the expected drift in robot trajectory cannot be accurately modeled to determine the appropriate loop closure search radius. The salient semantic features highlighted in the map can be used in a pre-matching step to improve loop closures. Moreover, there are costs associated with increasing D_r; while inflating the search radius could help to include the historical key-nodes that otherwise would not have been considered for loop closure, an expanded search space linearly increases the number of candidate key-nodes and subsequently leads to increase in the computational load associated with lidar scan matching that will impact the capability of the front-end to operate in real-time. Moreover, in self-similar and ambiguous environments, expanding the search space will increase the number spurious loop closures. If these are not successfully rejected, they can result in catastrophic distortions in the map.

Fig. 5.3-(a-b) show maps obtained in an indoor office environment by using the BGLC method where two search radii of 5m and 20m are used for loop closures. As illustrated with the blue lines between poses considered for loop closure, increasing the search

radius results in a dramatic increase in the number of attempted loop closures which not only increases the computational load associated with ICP-based lidar scan registration, but also increases the probability of spurious loop closures. This leads to catastrophic distortion of the map as shown in Fig. 5.3-(c).

Table 5.1 and Table 5.2 report the number of attempted and qualifying loop closures as the loop closure search radius is increased from 5m to 20m in both the indoor office environment and the Eagle mine. The tables show that doubling the search radius results in a dramatic increase in the number of attempted loop closures. This not only

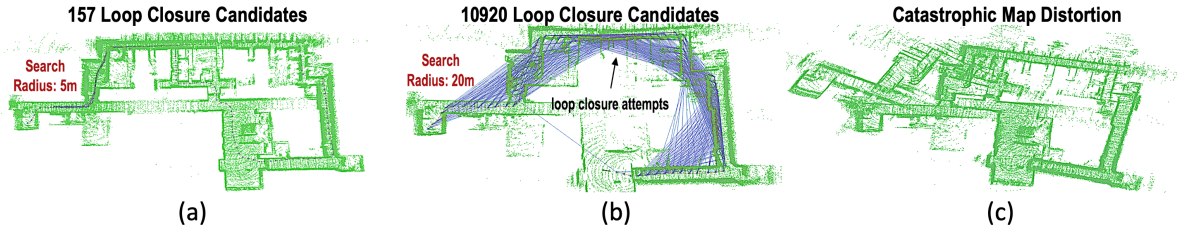

Fig. 5.3: (a) Expanding the loop closure search radius from 5m in (a) to 20m in (b) based on the BGLC method dramatically increases the number of attempted loop closures. The blue lines show all node pairs in the pose graph that are considered for loop closure. This in turn increases the probability of spurious loop closures that can result in distortions of the map as shown in (c).

Table 5.1: Indoor Office: number of attempted and qualifying loop closures based on ICP alignment error, as a function of loop closure search radius

LC Search Radius	Attempted LC	Qualifying LC	Percentage of Qualifying LC
5	323	21	6.501
10	3995	38	0.951
15	8480	42	0.495
20	10920	48	0.4395

Table 5.2: Eagle Mine: number of attempted and qualifying loop closures based on ICP alignment error, as a function of loop closure search radius

LC Search Radius	Attempted LC	Qualifying LC	Percentage of Qualifying LC
5	157	11	7.006
10	1705	20	1.173
15	2907	22	0.756
20	8728	36	0.412

leads to a significant computational cost that severely impacts the real-time operation of the front-end, but also leads to an increase of 80.95% and 81.81% in the number of spurious loop closures in the indoor office environment and the Eagle mine respectively.

In addition to challenges associated with the search for potential loop closures, with a poor initial alignment between two key-scans, the ICP algorithm may become trapped in a local minima [100] when the scene geometry does not constrain the ICP optimization sufficiently as discussed in Chapter 4. When the prior alignment of two key-scans is unknown, the ICP algorithm is initialized with an identity rotation and zero translation which could lead to a poor alignment and subsequently poor estimation of 3D motion between the two key-nodes as the ICP can fail to converge to the optimal solution. These vulnerabilities render the BGLC method unreliable and inaccurate, particularly in large-scale or long-duration navigation in perceptually-degraded environments.

To overcome these open challenges, in Section 5.2, a method is developed to improve performance and accuracy of loop closures in single and multi-robot SLAM systems without relying on any additional sensors. The proposed method is based on 2D projection of the 3D point clouds and extracting salient semantic features to enable a pre-matching step, during which the effective loop closure search radius is expanded as needed to account for the accumulated drift of the robot. In the next section, the proposed multi-stage loop closing method is described.

5.2 Semantic-Geometric Loop Closure (SGLC) Detection

The proposed method relies on semantic and geometric features extracted from lidar point cloud data to improve detection of loop closures in perceptually-degraded and ambiguous environments. Fig. 5.4 provides an overview of the method which consists of three main layers, namely pre-matching, geometric verification and outlier rejection.

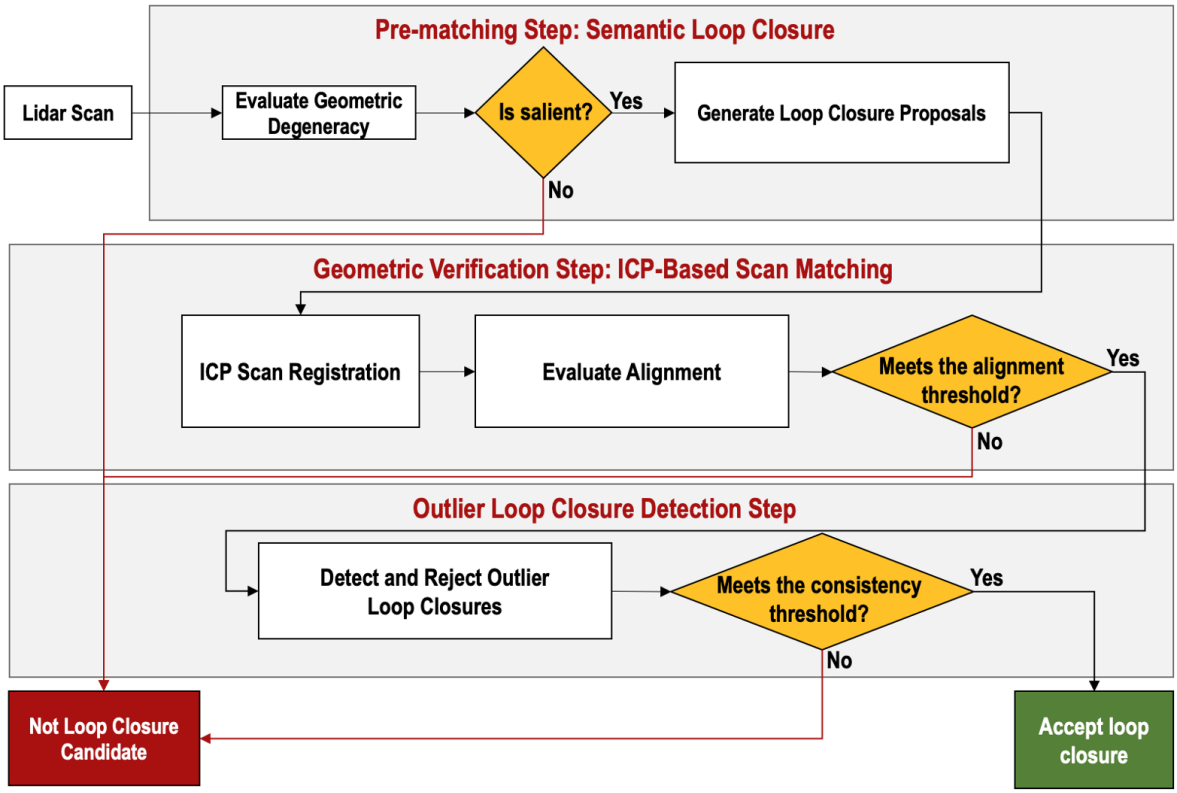

Fig. 5.4: An overview of the semantic-geometric loop closure detection method.

In the pre-matching step, 2D occupancy grid maps constructed from key-scans are used for an initial assessment and evaluation of potential loop closures over the robot trajectory. One of the key advantages of using a pre-matching step is that it enables a fast screening step independent of the estimated robot poses. In contrast to the BGLC method where the search space is constrained to a fixed radius centered at each robot pose, the pre-matching step is performed globally over the entire robot trajectory, enabling a *drift-resilient* loop closure detection framework. Potential loop closures that meet the similarity criteria are then passed to a geometric verification step, where the point clouds associated with candidate key-nodes are aligned using the ICP algorithm to further assess the quality of the loop closures. Finally, candidates that pass the geometric verification step undergo an outlier rejection step where the consistency of the detected loop closure with robot odometry and previous loop closures is evaluated

before adding the constrain to the pose graph. In the rest of this section each layer of the proposed multi-stage loop closure detection pipeline is described in detail.

5.2.1 2D occupancy grid maps

2D occupancy grid maps first introduced by A. Elfes in 1985 [101] are one of the most common types of maps used in robot navigation and path planning. Each 2D map represents robot's workspace as a discretized and fine grid over the continuous space of the robot's surrounding area. In probabilistic robotics, occupancy grid mapping addresses the problem of generating maps from noisy and uncertain lidar measurements with the assumption that the robot pose is known. The basic idea of the occupancy grid is to compute approximate posterior estimates of a map of the environment that is presented as an evenly spaced field of binary random variables each representing the presence of an obstacle at that location in the environment.

In this thesis, each occupancy grid map has 250×250 cells and corresponds to an area of 5 m $\times$ 5 m in the physical environment. In order to construct an occupancy grid map from a key-scan, the ROS 2D costmap tool [102] is used to obtain a slice of robot's surrounding 3D world from lidar point cloud data. This is achieved by filtering the point cloud to remove the points that comprise the ground plane, as well as the points that appear higher than the highest point on the robot. This is done because the purpose of the OGM is to determine if an obstacle would impede robot's motion. The filtered point cloud is then projected onto a XY plane to construct the occupancy grid map.

Due to the environmental and measurement noise in lidar scans, occupancy grid maps are represented as belief maps, where instead of using a binary value for occupancy, a probability is used to present the probability of occupancy of each cell. Each cell stores a corresponding occupancy belief $b_i(x, y),\ 0 \leq x, y < 250$, representing the estimated

probability that a static or stationary object is present in that cell location. A cell is considered as being in one of three possible states: free for $b_i(x, y) < 0.5$, unknown for $b_i(x, y) = 0.5$, and occupied for $b_i(x, y) > 0.5$. In order to obtain occupancy grid maps that can be used as binary map images, the belief map $b_i(x, y)$ is converted to a binary map $o_i(x, y)$ using

$$o_{\alpha_i}(x, y) = \begin{cases} 0 & \text{if } b_i(x, y) < 0.5 \\ 1 & \text{if } b_i(x, y) \geq 0.5 \end{cases}, \quad 1 \leq x \leq W, \;\; 1 \leq y \leq H. \tag{5.2}$$

Fig. 5.5, illustrates the occupancy grid map $o_{\alpha_i} = \{o_{\alpha_i}(x, y) : 1 \leq x \leq W, 1 \leq y \leq H\}$, constructed from the ith key-scan, where W is the number of cells in the width dimension and H is the number of cells in the height dimension.

5.2.2 Semantic saliency in occupancy grid maps

While loop closure detection is a crucial requirement in single- and multi-robot SLAM systems, it is equally crucial to avoid closing loops in ambiguous areas with high level of geometric degeneracy. Attempting loop closures in these areas can lead to detection of

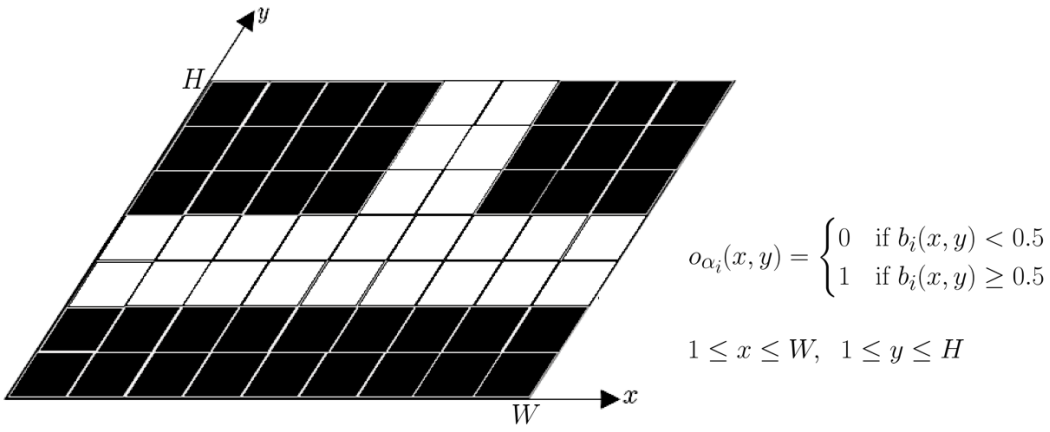

Fig. 5.5: Illustration of an occupancy grid map - white cells correspond to free space, and black cells correspond to occupied cells.

spurious or poor loop closures that can lead to catastrophic distortions of the constructed 3D maps. In an environment with high level of geometric degeneracy (e.g., a long and featureless tunnel with flat walls), the spatial configuration of the local scene captured in occupancy grid maps also lack sufficient distinctive features, a condition that we refer to it as *lack of semantic saliency*. Fig. 5.6 presents examples of occupancy grid maps constructed from lidar scans in an underground tunnel. While Fig. 5.6-(a) shows lack of semantic saliency that can lead to perceptual aliasing and data association ambiguity, Fig. 5.6-(b-c-d) show maps that contain salient semantic features sufficient for global localization in the pre-matching step.

By relying on the real-time determination of environment degeneracy developed in Chapter 4, the search for potential loop closures can be constrained to areas with sufficient levels of observability. The real-time determination of level of geometric degeneracy in the scene can be can be used to remove the occupancy grid map in Fig. 5.6-(a) from loop closure consideration where the scene is classified as degenerate. This leads to a significant reduction in data association ambiguity and perceptual aliasing in the pre-matching step.

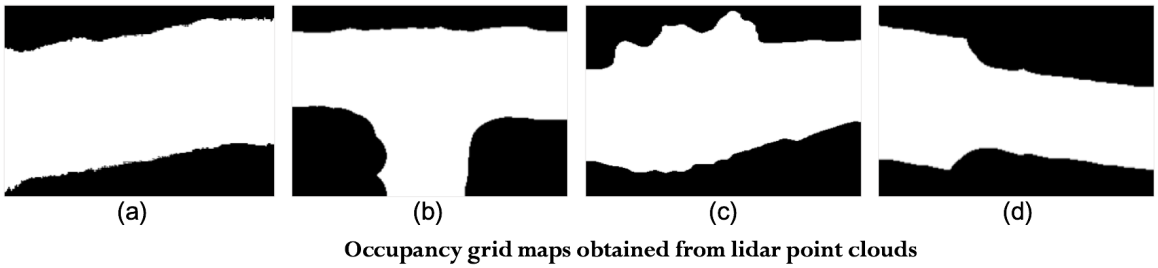

Fig. 5.6: (a) an ambiguous occupancy grid map obtained in a degenerate area with no salient features. (b), (c) and (d) are salient occupancy grid maps that contain distinctive landmarks.

5.2.3 Loop closure detection: pre-matching step

The pre-matching step can be considered as a place recognition process where a robot establishes global localization by registering the spatial configuration of its local scene to spatial configuration of the scenes visited in the past throughout its trajectory. Using 2D occupancy grid maps, this can be interpreted as a special instance of 2D image registration problem where occupancy grid maps can be viewed as 2D binary map images. Let o_{α_i} and o_{α_j} denote a pair of 2D occupancy grid maps (map images) obtained from key-scans $\bar{z}_{\alpha_i}$ and $\bar{z}_{\alpha_j}$ of robot α. Let $LC^s(\alpha_i, \alpha_j)$ denote the pre-matching step, where $LC^s(.,.)$ is a function that evaluates the level of similarity between a pair of map images based on a set of extracted features.

In order to register two map images, the set of N_f salient 2D features are extracted from o_{α_i} and o_{α_j}. In this thesis, the Oriented FAST and Rotated BRIEF (ORB) features by Ethan Rublee et al. [103] is used over other methods (i.e., SIFT, SURF) due to (i) outstanding speed and performance, (ii) resistance to image noise, (iii) and rotation invariance. Let $\mathcal{F}_{o_{\alpha_i}}$, and $\mathcal{F}_{o_{\alpha_j}}$ denote the set of extracted salient features from o_{α_i} and o_{α_j} respectively. We find the set of N_{corr} corresponding features by using the Fast Library for Approximate Nearest Neighbors (FLANN) presented by Chanop et al. [104]. FLANN is a fast local approximate nearest neighbors method that is commonly used to match keypoints found between corresponding images and to compute the set of corresponding feature points.

Finally, using the Random Sample Consensus (RANSAC) algorithm [79], a homography $\hat{\mathbf{M}}_{o_{\alpha_j}}^{o_{\alpha_i}}$ that best describes the 2D geometric transformation between the set of corresponding features is computed. The RANSAC algorithm used for estimating the homography determines the set of inlier features by iterating through three steps, (i) hypothesize, (ii) compute model parameters, and (iii) find consensus set, as outlined below:

- Hypothesize: select a minimal set of random samples (i.e., four feature pairs for computing homography),
- Compute model parameters: compute the homography $\hat{\mathbf{M}}$ using the elements of the minimal sample set,
- Find consensus set: check the elements of the entire corresponding feature set and find feature pairs that satisfy the inequality $\left\| \hat{\mathbf{M}} f_k^{o_{\alpha_i}}, f_k^{o_{\alpha_j}} < \epsilon_H \right\|$ for $k = 1 : N_{corr}$, where ϵ_H is a distance threshold between aligned features.

RANSAC terminates when the probability of finding a better ranked consensus set drops below a certain threshold. Finally, for the largest consensus set, the RANSAC algorithm finds the homography $\hat{\mathbf{M}}_{o_{\alpha_j}}^{o_{\alpha_i}}$ that describes the best estimated geometric transformation between two map images.

The objective of the pre-matching step is to identify the most likely loop closure candidates along robot's trajectory based on semantic similarity of the scenes. The correspondence confidence,

$$\zeta_{i,j} = \frac{N_{in}}{N_{corr}}, \tag{5.3}$$

which is the ratio of the number of inliers N_{in}, to the total number of corresponding points N_{corr}, is often used as a fitness measure in feature-based image registration. In this thesis, if the number of inliers $N_{in} \leq 20$, the map image is removed from loop closure consideration.

In perceptually-degraded environments, we can encounter situations where after finding the set of inliers using RANSAC, the correspondence confidence score is high even though the match is incorrect. In general, the theoretical breakdown point of all robust estimators, where there is no general guarantee of success in detection of true inliers, is

when the percentage of outliers is more than 50% [105]. As a result, based on the level of self-similarity and ambiguity between map images, the number of selected inliers and accuracy of the estimated homography can vary. In order to reduce perceptual aliasing and data association ambiguity, in this thesis the transformation confidence score is introduced to identify high values of correspondence confidence scores that are unreliable. Let $f_k^{o_{\alpha_i}}$ and $f_k^{o_{\alpha_j}}$ denote the kth pair of corresponding 2D features in map images o_{α_i} and o_{α_j}, where each feature is defined with its 2D image coordinate. Given the computed homography from RANSAC, the residual error between the kth pair of corresponding feature points is given by

$$r_k = \left\| \hat{\mathbf{M}} f_k^{o_{\alpha_i}} - f_k^{o_{\alpha_j}} \right\| \tag{5.4}$$

where r_k is the Euclidean distance between the kth pair of aligned corresponding points. Using (5.4), the mean squared error for the set of all N_k corresponding points is given by

$$\epsilon_{i,j} = \frac{1}{N_k} \sum_{k=1}^{N_k} \|r_k\|^2 . \tag{5.5}$$

Using the computed mean squared error $\epsilon_{i,j}$, we evaluate the quality of the computed homography between o_{α_i} and o_{α_j} by computing

$$\Lambda_{i,j} = \frac{1}{1 + \epsilon_{i,j}}, \tag{5.6}$$

where $\Lambda_{i,j}$ is the *transformation confidence* score which is close to 1.0 for a perfect match, and close to zero for a false match with a large number of outlier correspondences. This is used to reject cases which have a high correspondence confidence but also a high matching error by defining the *similarity confidence* metric denoted by $\Psi_{i,j}$, that is

obtained from the product of the correspondence and transformation confidence scores as given by

$$\Psi_{i,j} = \zeta_{i,j} \cdot \Lambda_{i,j}. \tag{5.7}$$

If the similarity confidence score is larger than a threshold, occupancy grid maps o_{α_i} and o_{α_j} are considered as a loop closure candidate.

Fig. 5.7 presents some representative examples of occupancy grid map matching where the set of corresponding and inlier features are visualized. Fig. 5.7-(a-b) present the correspondence and transformation confidence scores for two false image matches. In Fig. 5.7-(a) despite of having a large fraction of inlier features, the computed transformation confidence is too small indicating the poor quality of the match due to false set of inlier features. Fig. 5.7-(b) presents an opposite scenario where the number of inlier features is relatively small, while the computed transformation confidence score is large. This is mainly because most of the features are concentrated in patches at certain areas

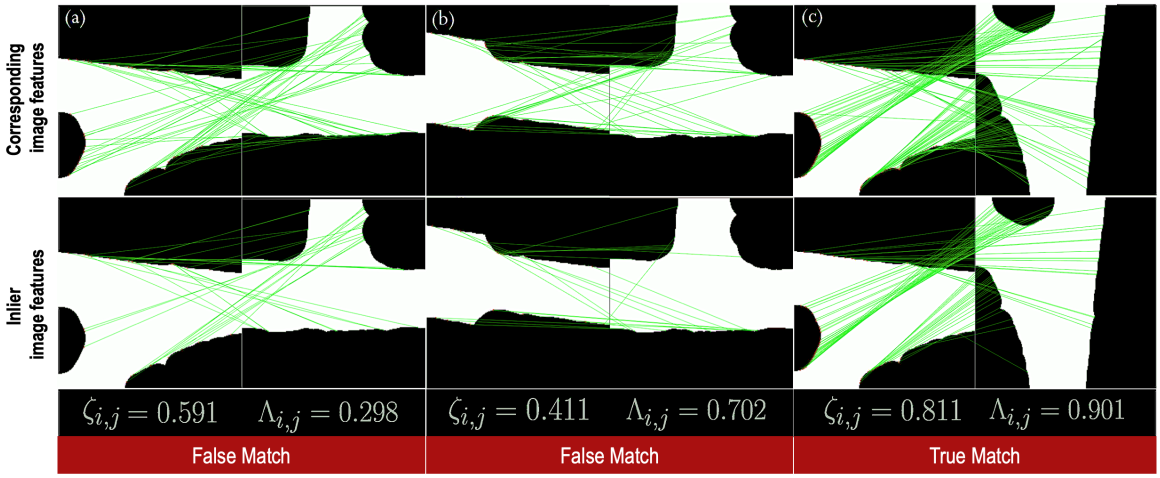

Fig. 5.7: The top and bottom rows present the set of corresponding and inlier features for an occupancy grid map that is matched against 3 salient occupancy grid maps. The correspondence and transformation confidence scores are computed for each pair. Only (c) shows a true positive match.

in the image resulting in a low alignment error. Fig. 5.7-(c) presents a true match where both correspondence and transformation confidence scores are large. As illustrated in the plots, while the individual correspondence and transformation confidence scores do not have enough discrimination power to determine a true loop closure, the product of both metrics can be used as a reliable similarity confidence metric to identify most similar matches.

Fig. 5.8 presents 2D scatter plots for correspondence and transformation confidence scores in an indoor office environment and an underground mine where 200 occupancy grid maps in each environment are evaluated for loop closure. On both plots, contours of similarity threshold values Ψ_{th} are shown. In the underground mine, a correspondence confidence level of $\zeta_{i,j} = 0.8$ would select the same loop closure candidates as $\Psi_{th} = 0.7$ and there are very few matches with $\zeta_{i,j} \geq 0.5$ that are rejected by Ψ_{th}. However, for the indoor office environment with self-similar office cubicles, a much larger number of candidate loop closures with $\zeta_{i,j} \geq 0.5$ are rejected by using the similarity confidence threshold Ψ_{th}. Fig. 5.9 and Fig. 5.10 show the characterization of the correspondence and transformation confidence scores using the occupancy grid maps obtained from key-scans in the indoor environment and the Eagle mine respectively. The plots show

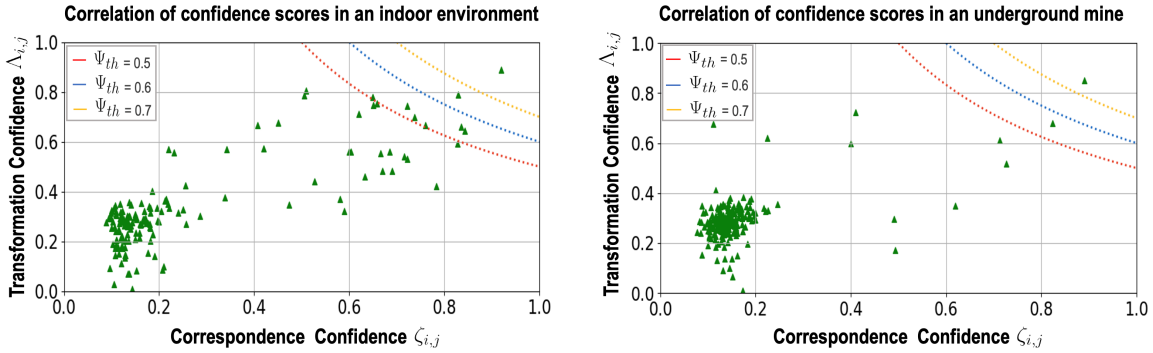

Fig. 5.8: 2D scatter plots showing the correspondence and transformation confidence scores for occupancy grid map matching in an indoor office environment and an underground mine.

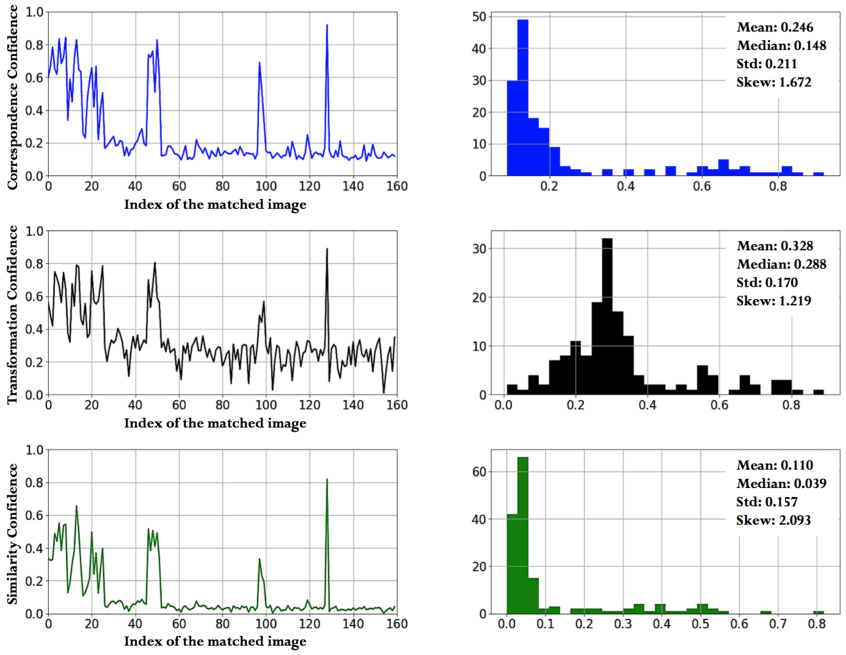

Fig. 5.9: Indoor office environment: plots and histograms of the correspondence, transformation and similarity confidence scores.

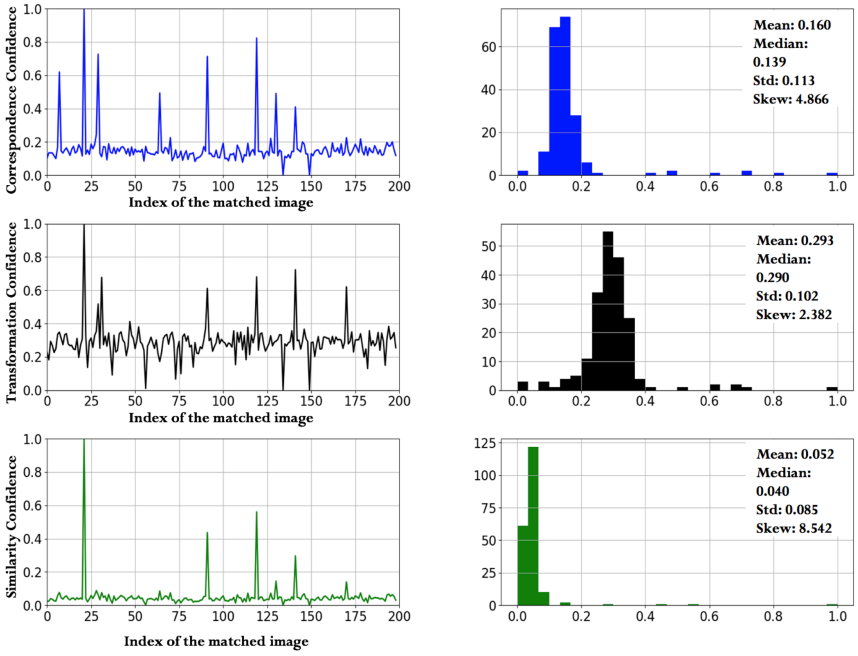

Fig. 5.10: Eagle mine: plots and histograms of the correspondence, transformation and similarity confidence scores.

the distribution of transformation and correspondence confidence scores, as well as the similarity scores computed for a set of occupancy grid maps constructed from the key-scans in these environments. While all three metrics are positively skewed, each of the correspondence confidence and transformation confidence metrics can individually lead to ambiguous results and perceptual aliasing as shown by the large standard deviation of computed scores. In contrast, the similarity confidence score has the highest positive skewness and the smallest standard deviation which makes it a more reliable metric with high discrimination power to determine the level of similarity between occupancy grid maps.

Using ROC analysis, we evaluate the sensitivity and specificity of the similarity confidence metric used in the pre-matching step in (5.7) in five underground environments. In each environment, a set of 100 salient occupancy grid maps are obtained from the key-scans, where 20 maps in each environment correspond to previously visited locations and represent true loop closures. Fig. 5.11 shows that loop closures can be reliably detected in various environments by relying on occupancy grid map matching where an average AUC = 0.756 is achieved in correctly identifying previously visited locations among all environments. Different threshold values are marked on each plot to show low

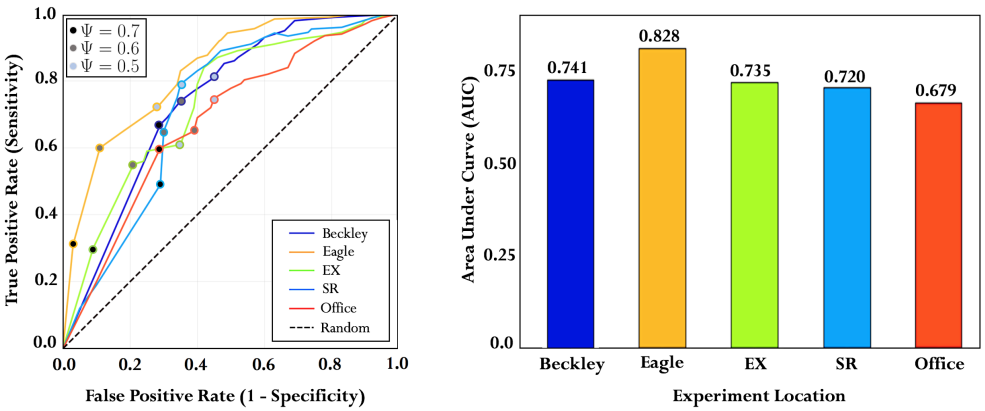

Fig. 5.11: The ROC curves and AUC for semantic loop closure detection in different environments.

sensitivity of semantic loop closures to threshold values across five different underground environments.

The indoor office represents a feature-rich environment with salient geometric structures. However, as presented in Fig. 5.11, the pre-matching step in this environment has a lower performance as compared to the underground mines. This is mainly due to perceptual aliasing and data association ambiguity that arises from the *self-similarity* of geometric structures in the indoor office environment. The presence of multiple office cubicles with identical geometric structure leads to larger number of false positive detections. While this increases the number of *candidate* loop closures, using a geometric verification step and a back-end equipped with an outlier loop closure detection capability as depicted in Fig. 5.4, each loop closure candidate is verified in terms of accuracy and consistency before being added to the pose graph.

5.2.4 Loop closure detection: geometric verification step

Let α_i, α_j and $\hat{\mathbf{M}}_{o_{\alpha_j}}^{o_{\alpha_i}}$ be the candidate loop closure key-nodes in the pose graph and the computed homography obtained from the pre-matching step. Using the key-scans associated with the candidate key-nodes, as presented in Fig. 5.4, a geometric verification step $LC^g(LC^s(\alpha_i, \alpha_j))$ is used to verify the quality of a loop closure candidate using the ICP-based scan registration as shown in (3.3), where $LC^g(.,.)$ denotes the geometric verification step based on the ICP algorithm.

The performance of the ICP algorithm relies heavily on the quality of initialization; with a poor initial guess the algorithm is susceptible to local minima, especially if the actual 3D motion between two lidar scans is large. In order to improve convergence of the ICP algorithm to the optimal solution in the geometric verification step, a semantic assisted initialization used by relying on the estimated 2D rotation matrix obtained from the

computed homography $\hat{\mathbf{M}}_{o_{\alpha_j}}^{o_{\alpha_i}}$ in the pre-matching step. Using this rotation information, a 3D transformation matrix with a Z-axis rotation and zero translation is constructed for seeding the ICP algorithm. This can be interpreted as a two-stage optimization process where the objective is to refine the computed geometric transformation obtained from semantic loop closure detection step to find the best estimate of the relative 3D motion between two key-scans. After aligning the point clouds, the convergence criteria is evaluated using the alignment error $\mathcal{E}(.)$ from (3.2) before sending the loop closure constraint to the SLAM back-end for outlier rejection verification.

5.2.4.1 Analysis of computational complexity

In this section, the computational complexity introduced by the pre-matching step is characterized by evaluating the execution time of the semantic loop closure detection step using datasets of different sizes. In order to characterize the computational load introduced by adding the pre-matching step to the loop closure detection process, the computation time required to perform occupancy grid map matching on multiple datasets is measured. For each dataset, 10 random salient occupancy grid maps are used as query map images. Using the method described in Section 5.2, each query map image is registered to all map images in a dataset and a similarity confidence score is computed. The total computational time required to iterate over all images in a dataset is recorded and shown using the box plots presented in Fig. 5.12. The average computational time required to detect a semantic loop closure shows a logarithmic growth as the size of the dataset grows. This is mainly due to the fact that semantic loop closure is only attempted for a subset of images in the dataset where the number corresponding features between the query image and a map image in the dataset is larger than the threshold $corr_{th}$.

As presented in Chapter 5, a main drawback of the commonly used BGLC method is

that the search for loop closures is constrained to a fixed radius centered at each robot pose. If the search space is expanded to include all the nodes in a pose graph, each new key-scan is registered to all previously obtained key-scans to find loop closures. This can become increasingly computationally expensive, especially in large-scale or long-duration navigation. Considering the significant number of points in each lidar scan, this can result in the quadratic computational complexity $O(n^2)$ [106] of the ICP algorithm, incurring the BGLC method a prohibitive computational complexity cost of $O(n^3)$ in detection of loop closures for n nodes in the pose graph.

In the proposed SGLC method, the pre-matching step enables the expansion of the loop closure search space to all the nodes in the pose graph without a significant increase in computation or data association ambiguity. The pre-matching step relies on a pair-wise occupancy grid map registration process with $O(n)$ complexity [107]. While determination of geometric degeneracy helps to reduce the number candidates by constraining the search to areas with highest observability, in a feature-rich environment all constructed occupancy grid maps will have sufficient semantic saliency to be considered in the pre-matching step. In this case the pre-matching step will incur a computational complexity cost of $O(n^2)$ for the set of all n binary map images. By relying on the proposed metrics

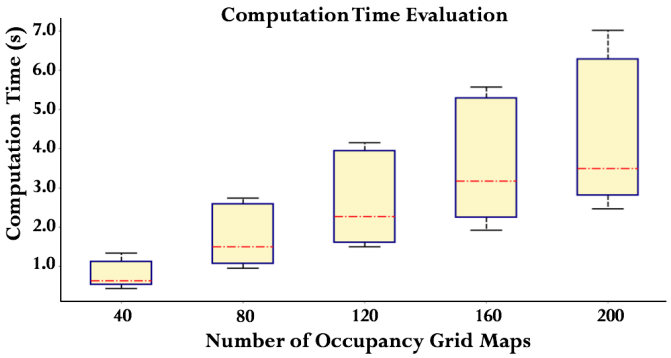

Fig. 5.12: Semantic loop closure detection computational time. Each box comprises the computation time values ranging from the first to the third quartile. The median is indicated by the dashed red horizontal bar. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range.

to determine level of geometric degeneracy, the semantic loop closure is only attempted at locations where the environment is fully observable. This underscores the importance of relying on *salient* occupancy grid maps for loop closure detection in order to reduce the required computation time, as well as reducing the perceptual aliasing associated with matching ambiguous occupancy grid maps. In feature-rich environments with a large number of qualifying nodes in the graph, the pre-matching step can be executed on a separate thread to avoid affecting the real-time performance of the front-end. Furthermore, once a loop is successfully closed and the drift in the computed robot trajectory is minimized, the subsequent loop closures can be delayed until a minimum number of new key-nodes are instantiated in the pose graph.

5.2.5 Loop closure detection: outlier rejection step

The local back-end receives the odometry measurements from the front-end and periodically instantiates new key-nodes in the graph after a minimum odometric displacement has occurred. Since pose graph optimization relying on least square optimization methods is not robust against outliers, the back-end has to rely on an outlier loop closure detection module to prevent the optimization from producing incorrect solutions when the front-end creates erroneous loop closure constraints due to errors in the underlying data association.

The outlier loop closure detection is based on the Pairwise Consistent Measurement Set Maximization (PCM) method presented in [4,108] which has shown high reliability and performance in rejection of outlier intra- and inter-robot loop closures. Once a loop closure is detected, the quality of the loop closure is evaluated both in terms of pairwise consistency against other loop closures as in the method presented by Mangelson et al. [109], and against the odometry edges. As illustrated in Fig. 5.13, in PCM the basic

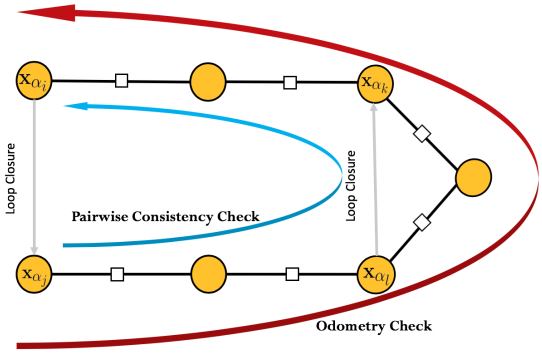

Fig. 5.13: Outlier loop closure rejection based on odometry and pairwise consistency check.

observation is that, in the absence of noise, relative pose measurements should compose to the identity along cycles in the graph [110], where cycles are defined as complete loops in the pose graph.

5.2.5.1 Odometry consistency check

Assume that the front-end produces a putative loop closure measurement $\mathbf{u}_{\alpha_j}^{\alpha_i}$, between key-nodes α_i, α_j. Then, the measurements along the cycle formed by the loop closure $\mathbf{u}_{\alpha_j}^{\alpha_i}$, and the odometry edges must compose to the identity as given by

$$\mathbf{T}_{\alpha_i,\alpha_j}^{err} \doteq \mathbf{u}_{\alpha_j}^{\alpha_i} \cdot \mathbf{T}_{\alpha_j,\alpha_i}^{odom}, \tag{5.8}$$

where $\mathbf{T}_{\alpha_i,\alpha_j}^{err}$ is the relative transformation error obtained by chaining all odometry edges $\mathbf{T}_{\alpha_j,\alpha_i}^{odom}$ from key-node α_j to key-node α_i, and the loop closure edge $\mathbf{u}_{\alpha_j}^{\alpha_i}$. The back-end rejects loop closures if the computed relative transformation error exceeds a set rotation threshold θ_{PCM-th} or a translation threshold t_{PCM-th}.

5.2.5.2 Pairwise consistency check

Similarly on the base station, the relative transformation error $\mathbf{T}^{err}_{\alpha_i,\beta_j}$ can be computed by creating a loop between two inter-robot loop closures, or between an inter-robot loop closure and robots' initial positions, assuming the initial positions of the robots in the world coordinate system are known. In order to check if $\mathbf{u}^{\alpha_i}_{\alpha_j}$ is consistent with a previous loop closure $\mathbf{u}^{\alpha_l}_{\alpha_k}$, the loop error is computed as:

$$\mathbf{T}^{err}_{\alpha_i,\alpha_j,\alpha_k,\alpha_l} \doteq \mathbf{u}^{\alpha_i}_{\alpha_j} \cdot \mathbf{T}^{odom}_{\alpha_j,\alpha_l}, \cdot \mathbf{u}^{\alpha_l}_{\alpha_k} \cdot \mathbf{T}^{odom}_{\alpha_k,\alpha_i} \tag{5.9}$$

To keep track of pairwise consistency of pairwise loop closures, an adjacency matrix is used as in [109]. If the detected loop closure is an outlier, the computed relative transformation error exceeds a set rotation threshold θ_{PCM-th}, or a translation threshold t_{PCM-th}, and the loop closure is rejected as an outlier. After rejecting the outlier loop closures, a loop closure edge $\mathbf{u}^{\alpha_i}_{\alpha_j}$ is added to the local pose graph.

Table 5.3 and Table 5.4 present the number of detected and qualifying loop closures by relying on the PCM-based outlier rejection module, using different rejection thresholds for the Eagle mine and the indoor office environment. The result show relaxing the loop closure criteria will significantly increase the number of qualifying loop closures. In experiments presented in this thesis loop closures with rotation error larger than 5° or translation error larger than 0.005m are rejected as outliers. Finally, once an intra- or inter-robot loop closure is confirmed and a new constraint is added between two key-nodes, the pose graph is optimized using iterative nonlinear optimization methods, e.g. the Gauss-Newton method, to estimate all robot poses and positions of artifacts in the map.

Loop closures that pass the loop closure rejection criteria are added to the pose graph using the SLAM back-end. The back-end is implemented using the Georgia Tech Smooth-

Table 5.3: Eagle Mine: number of attempted and qualifying loop closures using PCM

LC Search Radius	Attempted LC	Qualifying LC 5°, 0.005 m	Qualifying LC 10°, 0.010 m	Qualifying LC 15°, 0.15 m
5	157	11	39	57
10	1705	20	53	78
15	2907	22	71	88
20	8728	36	85	97

Table 5.4: Indoor Office: number of attempted and qualifying loop closures using PCM

LC Search Radius	Attempted LC	Qualifying LC 5°, 0.005 m	Qualifying LC 10°, 0.010 m	Qualifying LC 15°, 0.15 m
5	323	21	70	79
10	3995	38	75	88
15	8480	42	83	103
20	10920	48	98	114

Algorithm 1 Semantic-Geometric Loop Closure (SGLC) Detection

Input : a query lidar scan z_i
Input : a set of previously obtained lidar scans Z
Output: a loop closure detection

procedure FindLoopClosure(Z, z_i)
  if $(log(\kappa_i) \leq \kappa_{th})$ **then**
    $o_i \leftarrow$ ConstructOccupancyGridMap(z_i)
    for $(\forall z_j \in Z)$ **do**
      if $(log(\kappa_j) \leq \kappa_{th})$ **then**
        $o_j \leftarrow$ ConstructOccupancyGridMap(z_j)
        if (SemanticLoopClosure(o_i, o_j)) **then**
          return $\mathbf{M}_j^i$
          if (GeometricVerification($z_i, z_j, \mathbf{M}_j^i$)) **then**
            return $\mathbf{u}_j^i$
            if NotOutlierLoopClosure($\mathbf{u}_j^i$) **then**
              return $\mathbf{U} \leftarrow \mathbf{U} \cup \mathbf{u}_j^i$
            end if
          end if
        end if
      end if
    end for
  end if
end procedure

ing and Mapping (GTSAM) [111] library that implements smoothing and mapping using factor graphs and Bayes networks as the underlying computing paradigm. Algorithm 1 summarizes the multi-stage loop closure detection process.

5.3 Loop Closure Detection: Multi-Robot Systems

The field of cooperative multi-robot systems has been subject of considerable research efforts in the last decade. Multi-robot systems have a broad application aspect thanks to advantages of distributed sensing and actuation, and can perform tasks with increased performance, redundancy and fault tolerance as compared to single-robot systems. Some examples of their broad range of applications include public and private service robots, autonomous air and ground transportation systems, and robotic search, rescue and surveying of hazardous and unknown environments in disaster response scenarios. Moreover, planetary applications like exploration of lunar and Martian surfaces or lava tubes can benefit largely from the use of multi-robot systems.

In order to show that the proposed degeneracy-aware and drift-resilient loop closure detection method can be valuable for improving inter-robot loop closures and map merging in multi-robot systems, in this dissertation a centralized system is adopted that includes a base station responsible for merging the individual trajectories created by a group of spatially or temporally-separated robots exploring an unknown and perceptually-degraded environment. An overview of the proposed multi-robot SLAM architecture is presented in Fig. 5.14. By relying on the onboard 3D lidar scanner each robot performs single-robot SLAM system to estimate its trajectory and the map of the environment. The local back-end is responsible for graph construction, rejection of spurious intra-robot loop closures, and pose graph optimization to obtain the best estimate of the robot trajectory and the 3D map of the environment. The result of the local front-end

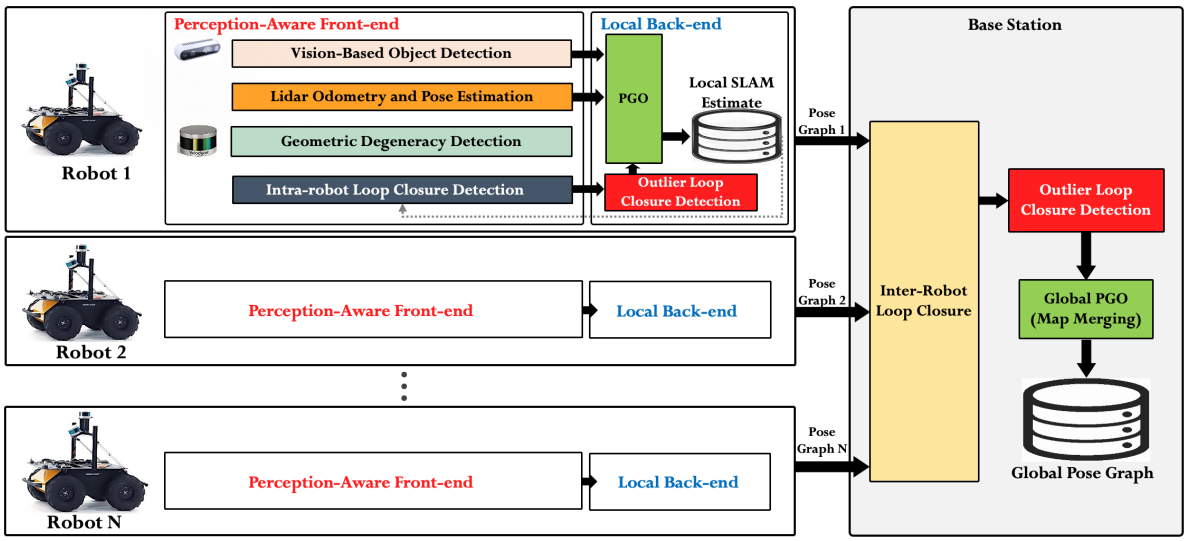

Fig. 5.14: Centralized multi-robot collaborative mapping architecture.

and back-end is an estimate of the robot trajectory modeled as a set of poses and artifact positions, as well as a set of obtained lidar point clouds. Each robot builds an explicit local map of the environment by projecting the point clouds associate with the nodes in the pose graph into a common world coordinate system $\mathcal{W}$.

When a robot is within the communication range, it communicates its local pose graph to the base station. Assuming the initial positions of robots are known, and there is at least a partial overlap between robot trajectories, map merging is performed on the base station by finding the correspondences between the pose graphs. This is achieved using the same multi-stage loop closure detection process used in the single-robot scenario, where using a degeneracy-aware pre-matching step the search for loop closures is constrained to the most observable areas in the local pose graphs to identify putative loop closures. Upon determination of inter-robot loop closure candidates, each candidate undergoes geometric verification and outlier rejection steps before performing a global pose graph optimization to find the most probable configuration of the nodes in the merged pose graphs given the set of all intra- and inter-robot constraints.

5.4 Experimental Results and Analysis

This section presents performance and computational analysis of the proposed semantic-geometric loop closure (SGLC) detection method in a variety of complex underground environments listed in Table 5.5. Most SLAM systems have inadequate performance when deployed in these unknown and GPS-denied environments; sensors must operate in off-nominal conditions (poor illumination or lack-thereof, dust, water puddles and non-Lambertian surfaces) which render visual-SLAM approaches unreliable. Uneven and slippery terrains make wheel odometry inaccurate, while long, featureless corridors make lidar-based mapping prone to drift; finally, perceptual aliasing, the presence of many similar-looking corridors and intersections, and sparsity of salient geometric structures induces spurious loop closures that can degrade the mapping results. Fig. 5.15 presents some examples of extreme and challenging underground environments used for evaluation and performance analysis of the proposed loop closure detection method. This section provides quantitative and qualitative comparison of the localization and mapping results against the basic geometric loop closure (BGLC) detection method, as well as the LeGO-LOAM [6] method which is a state-of-the-art SLAM algorithm with loop closure detection capability.

As presented in Fig. 5.14, in this thesis a centralized multi-robot SLAM architecture is used in real-world experiments. The architecture discussed in this paper is developed in the context of DARPA Subterranean Challenge where the main objective of this

Table 5.5: List of the explored underground mines.

Name of the mine	Autonomously Traversed Distance	Type of mine	Location
Arch Pocahontas Mine	1100 m	Coal Mine	Beckley, WV
Beckley Exhibition Mine	1000 m	Coal Mine	Beckley, WV
Bruceton Safety Research Mine	1400 m	Coal Mine	Pittsburgh, PA
Bruceton Experimental Mine	700 m	Coal Mine	Pittsburgh, PA
Highland Mine	1400 m	Coal Mine	Logan, WV
Eagle Mine	500 m	Gold Mine	Julian, CA

Fig. 5.15: Examples of the perceptually-degraded and extreme underground environments explored by autonomous ground robots.

world-wide robotic challenge is to explore and map unknown and extreme subterranean environments (Fig. 5.15) using single- or multi-robot systems. In this centralized SLAM architecture, a base station is responsible for receiving the local pose graphs from individual robots and merging them together by detecting the correspondences between the maps to build a consistent global map of the unknown environments. The base station that is an Intel Hades Canyon NUC8i7HVKVA (4×1.9 GHz, 32 GB RAM) does not communicate the constructed global maps back to the robots and thus, a real-time performance is not strictly required as map alignment and merging can be performed in an offline post-processing step. The constructed global maps can be used for future reliable navigation of robots in the mapped environments. As presented in Fig. 5.16, the multi-stage loop closure detection method developed in this thesis is a key component of the global inference SLAM front-end module of the Networked Belief-aware Perceptual Autonomy (NeBula) framework. NeBula is an architecture motivated by autonomous exploration of extreme surfaces and subsurface terrains on earth and planetary bodies that is currently being developed at NASA's Jet Propulsion Laboratory. The main focus of NeBula is to provide computationally tractable methods to predict and assess various

outcomes and risks in uncertain settings. These methods subsequently enable reliable, coordinated multi-robot exploration of unknown and hard-to-access terrain.

The perception component of NeBula consists of odometry, odometry multiplexer, and local mapping modules [112]. The global inference builds a global map, and learns high-level information from the environment [4, 113]. The belief manager block constructs and maintains the robot's model of the environment [114–116]. The mission planning switches between various behaviors, such as stair-climbing and communication recovery, and global planning guides the coverage behavior [117, 118]. The local planner assesses the traversability risk and plans safe paths for the robots [119–121]. The communication module is responsible for enabling data exchange between multiple robots and a base station. The belief prediction module is a critical component in the NeBula architecture that enables perception-aware and uncertainty-aware planning. This module allows the planner to take perceptual capability into account and helps reduce the risk by increasing the accuracy of the world representation. For more information about NeBula autonomy system, the readers are referred to [122].

In experiments presented in this thesis, robots are Husky-A200 series that rely on a VLP-16 Puck Lite lidar scanner [10] and an Intel NUC 7i7DNBE (4×1.9 GHz, 32 GB

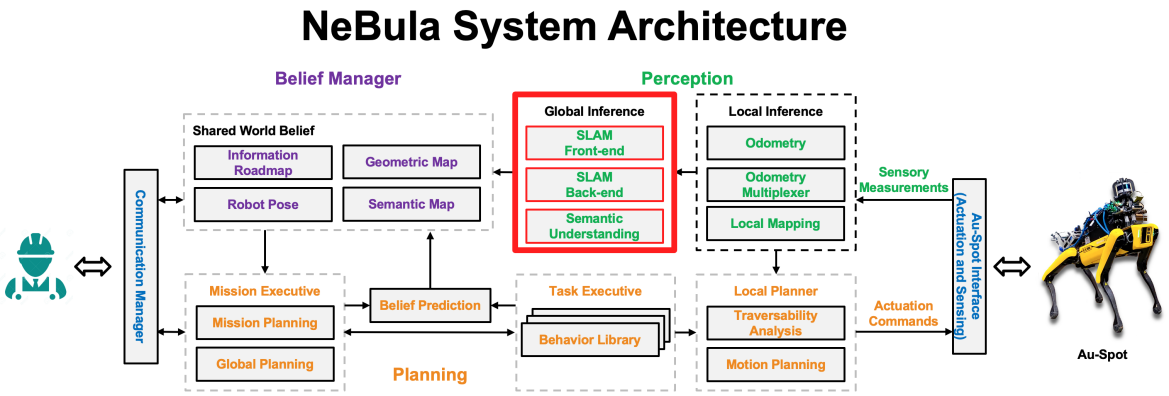

Fig. 5.16: Examples of the perceptually-degraded and extreme underground environments explored by autonomous ground robots.

RAM) processor for performing simultaneous localization and mapping. By relying on the onboard Intel RealSense D435 RGB-D camera, each robot uses vision-based object detection based on the You Only Look Once (YOLO) [123] object detection algorithm to detect, classify and localize objects of interest in the environment. When a robot is within the wireless communication range, it communicates its trajectory and the constructed map with the base station.

As described earlier in Section 1.3, obtaining the ground truth data in the explored large-scale underground environments is a challenging task. In this thesis, the method presented in LAMP [4] is used to obtain a proxy for the ground truth trajectories by enforcing the ground truth locations of the known objects and fiducial markers in the best pose graph of each robot and use the resulting optimized trajectory as ground truth. Fig. 5.17 shows a ground truth map of the Bruceton Safety Research mine.

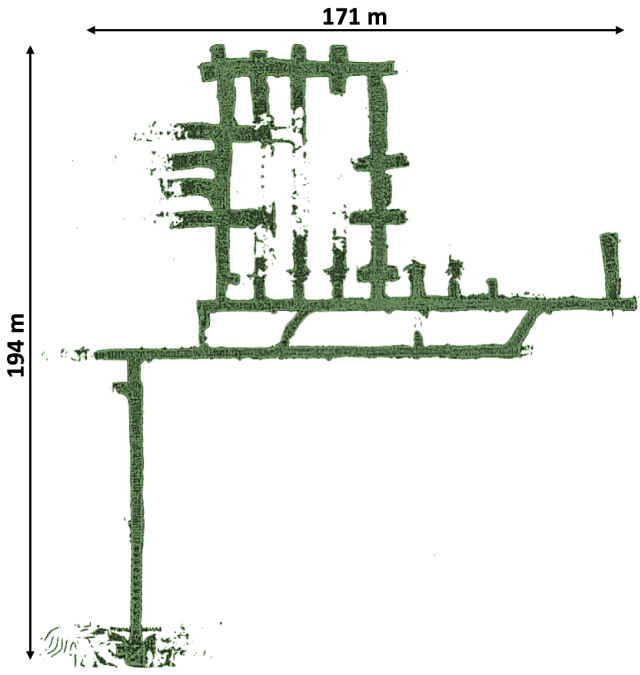

Fig. 5.17: A partial ground truth map of Bruceton Safety Research mine, obtained by enforcing the ground truth locations of the known objects and fiducial markers in the best pose graph of a robot.

Fig. 5.18 and Fig. 5.19 show examples of constructed maps in two underground mines. At the Beckley Coal mine a robot autonomously traverses more than 1km in the mine and returns to the start location to close the loop. Fig. 5.18-(a) shows the constructed map in an open-loop scenario where no loop closure detection capability is used. Fig. 5.18-(b-c) show the maps obtained in two trials using the BGLC method where a fixed search radius of 10m is used to search for loop closures. The results show many loop closure opportunities are missed leading to the dramatic distortion of the maps. Fig. 5.18-(d) shows the map obtained using the proposed SGLC method which is pose-invariant, resulting in successful detection of loop closures over the entire robot trajectory to obtain a more consistent estimation of the map of the environment.

Fig. 5.19 presents a multi-robot collaborative mapping scenario where local maps obtained by a team of two robots deployed in the Eagle mine are merged on a base station.

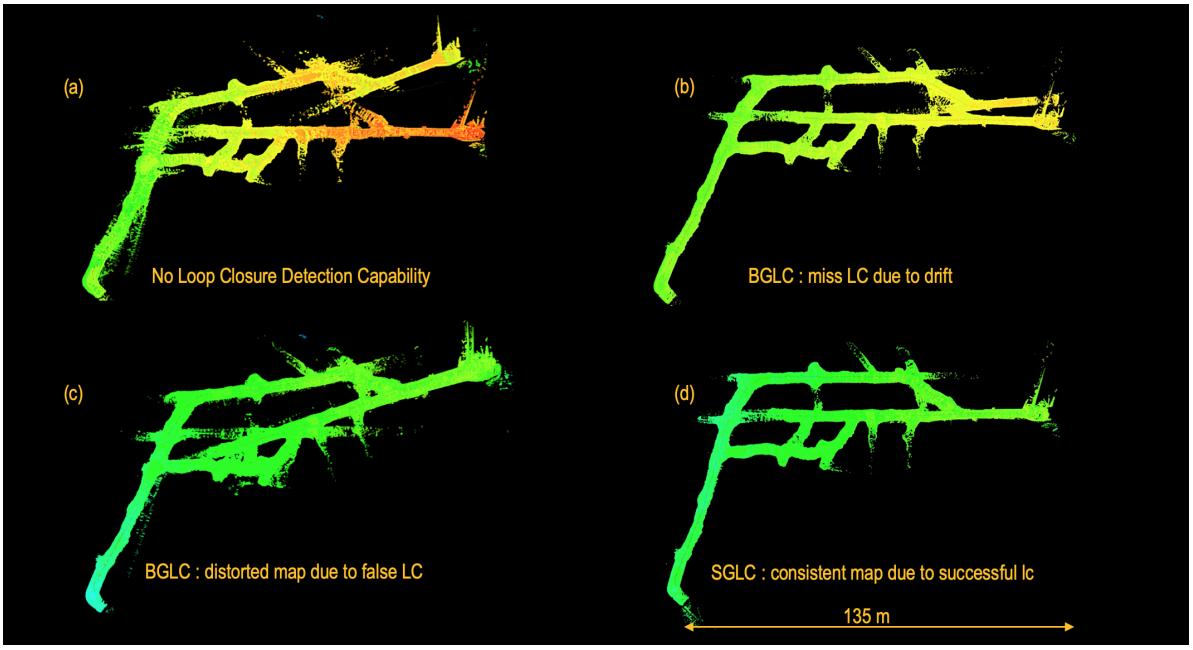

Fig. 5.18: Top-down view of the 3D Map of the Beckley coal mine, Beckley, WV. (a) mapping with no loop closure detection capability. (b) and (c) using the BGLC method maps are distorted due to spurious and missed loop closure opportunities. (d) constructed map using the proposed SGLC method.

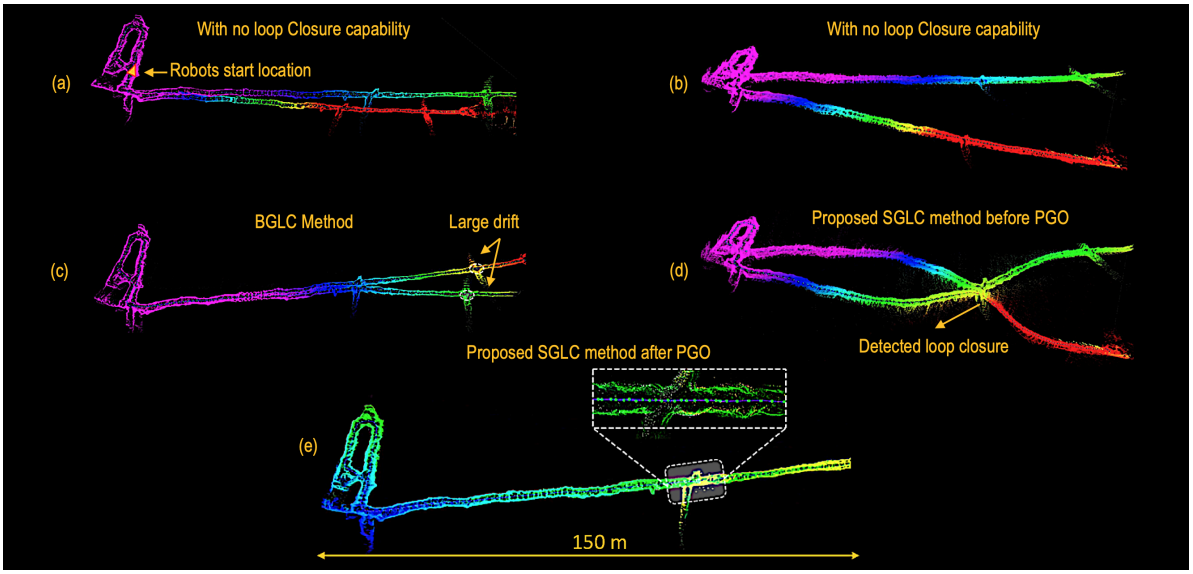

Fig. 5.19: Map of the Eagle Mine, Julian, CA obtained on a base station. (a) and (b) present the top-down and side views of the 3D map obtained on the base station with no loop closure detection capability. (c) mapping result using the BGLC method. (d) map before performing PGO - a loop closure is detected between two nodes in the pose graphs using the proposed SGLC method. (e) global map after performing PGO.

Both robots with known initial poses in the world coordinate system autonomously navigate 500m in the tunnels of the mine. Fig. 5.19 present the top (a) and side view (b) of the maps obtained on the base station in an open loop scenario without using any loop closure detection capability. As the error in robot trajectories accumulates over time, the maps start to drift unbounded in the absence of loop closure detections. Using the BGLC method, the search for inter-robot loop closures is performed on the base station where for every key-node in one pose graph, key-nodes in the other graph that lie inside the search radius of 10m are considered for loop closure. For each candidate, the base station performs an ICP-based scan registration using (3.3), and evaluates the alignment error of point clouds. As presented in Fig. 5.19-(c), while the BGLC method performs well when the drift in robot trajectories is small, it misses many loop closure opportunities when the trajectories drift apart due to accumulation of noisy odometric estimates. Fig. 5.19-(d) shows a successful loop closure using the proposed

SGLC method before pose graph optimization is performed. Using the salient semantic features (i.e., a T-junction in the tunnel) an inter-robot loop closure is detected and the diverged trajectories are joined again. Fig. 5.19-(e), shows the drift is minimized and a more consistent map of the environment in obtained after performing pose graph optimization.

Fig. 5.20 presents the maps obtained by a single robot exploring the indoor office environment, where the robot returns to the initial location to close the loop at the end of the experiment. Fig. 5.20-(a) shows the final map obtained by relying on the BGLC method. As the location of the search space relies on the estimated robot poses, multiple loop closure opportunities are missed due to significant drift in the estimated robot trajectory. Fig. 5.20-(b) presents the maps obtained by relying on semantic-geometric loop closure detection method where the search for loop closures is performed over the entire robot trajectory independent of the estimated robot poses. The detection of successful loop closures leads to significant reduction of drift and a more consistent global map of the environment.

Fig. 5.21 reports examples of multi-robot mapping for a team of two robots deployed in the Bruceton Experimental Mine and Safety Research Coal Mine during the Tunnel Circuit of DARPA Subterranean Challenge [124], in August of 2019. Using the BGLC method, large drifts are visible in both maps as the base station fails to merge the maps due to many missed loop closure opportunities when the trajectories start to drift apart. The maps obtained using the proposed SGLC method show more consistent representation of the explored mines as loop closures are detected more frequently.

At the Tunnel Circuit competition of the DARPA Subterranean challenge, a variety of objects (e.g., backpack, fire extinguisher, drill, survivor, and a cellphone) were placed at unknown locations in the environment. The autonomous robots deployed in the environment were used to detect and localize these objects in the environment by relying

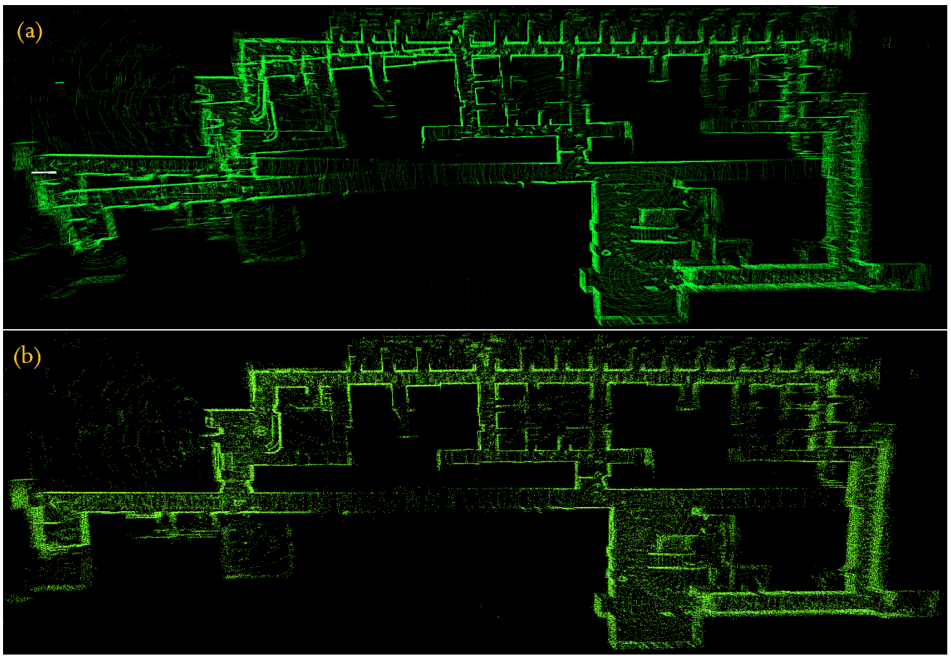

Fig. 5.20: (a) Intra-robot loop closure detection in an indoor office environment. (a) Final map using geometric loop closure detection. (b) Final map using semantic-geometric loop closure detection.

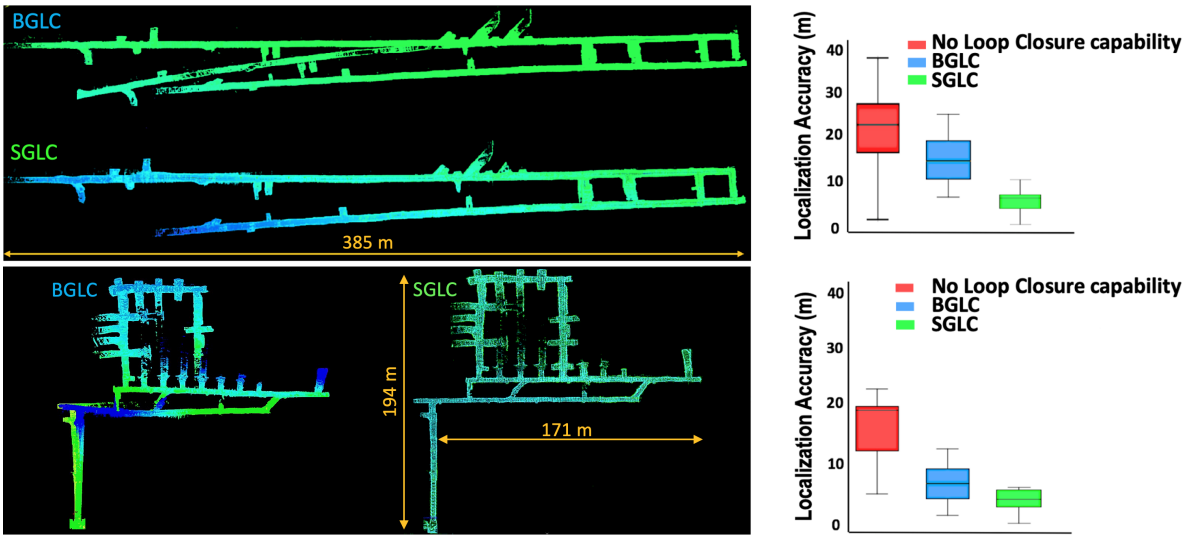

Fig. 5.21: Maps of the Bruceton Experimental and Safety Research mines constructed using the BGLC and proposed SGLC methods. The box plots report the artifact localization errors in the final maps, without loop closure detection capability, as well as the BGLC and the proposed SGLC methods.

on the onboard RGB-D camera and the lidar-based SLAM. The estimated location of the detected objects is used to provide a quantitative evaluation of localization and mapping accuracy. The box plots in Fig. 5.21 report a quantitative evaluation of localization accuracy using no loop closure detection capability, the BGLC method, and the proposed SGLC method by comparing the estimated location of detected artifacts against the ground truth data provided by DARPA. With no loop closure detection capability, the robot pose uncertainty and location uncertainty of detected objects grows larger as the lidar odometry errors accumulate. By using the BGLC method, several missed loop closure opportunities lead to accumulation of significant drift in robot trajectories, which manifests itself in large artifact localization errors. Using the proposed SGLC method, loop closures are consistently detected as the robot navigates the unknown environment, which results in a significantly better localization accuracy as compared to the BGLC method.

Fig. 5.22 and Fig. 5.23 provide comparison of the 3D maps obtained using the LeGO-LAOM, BGLC and proposed SGLC method in five perceptually-degraded and large-scale underground environments. While in the released LeGO-LOAM software the default search radius for loop closures is set to 7m, in our experiments the search radius was increased to 10m for both LeGO-LOAM and the BGLC methods in order to improve loop closure detections in the presence of large drift. The results show both LeGO-LOAM and BGLC methods initially perform well as long as the drift in robot trajectories is small. However, both methods miss several loop closure opportunities in the presence of large drift. This highlights one of the main drawbacks of pose-dependent loop closure detection methods, especially in large-scale and perceptually-degraded environments where the accumulation of odometry errors can lead to significant drift. The maps show our proposed SGLC method consistently outperforms the LeGO-LOAM and BGLC methods as it does not rely on estimated robot poses to identify loop closure

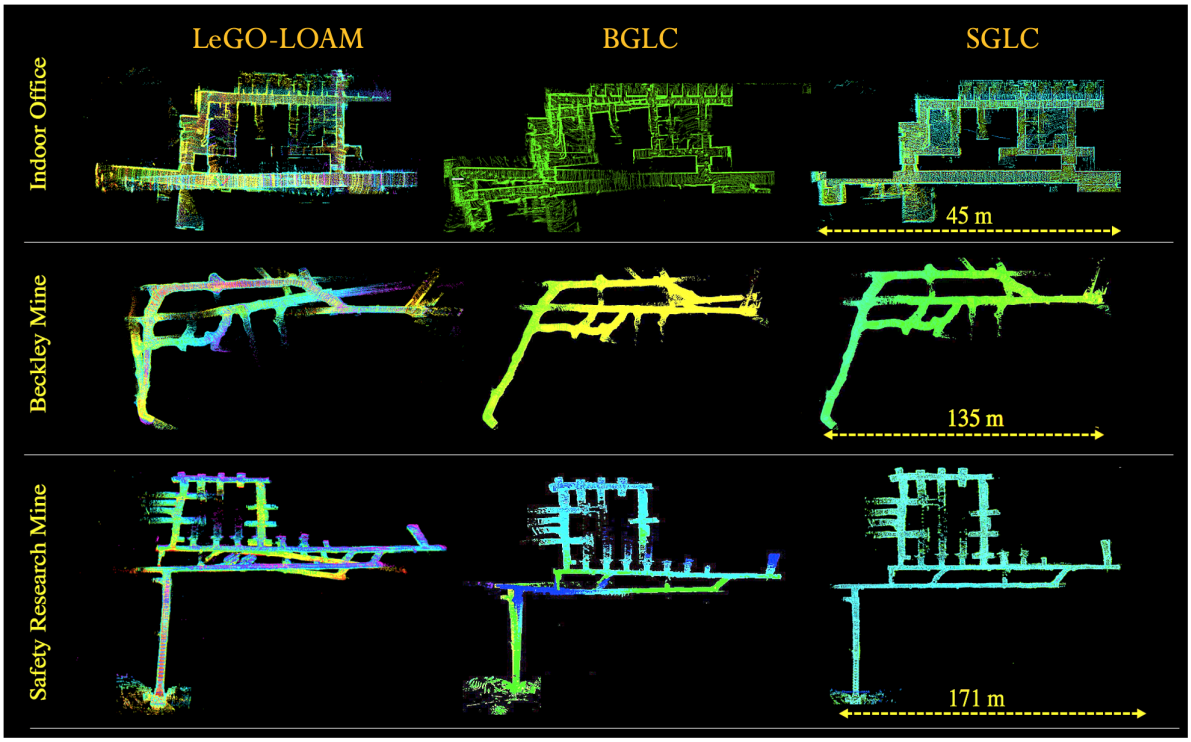

Fig. 5.22: Maps of the indoor office, Beckley mine, and Safety Research mine obtained using the LeGO-LOAM, BGLC and proposed SGLC methods.

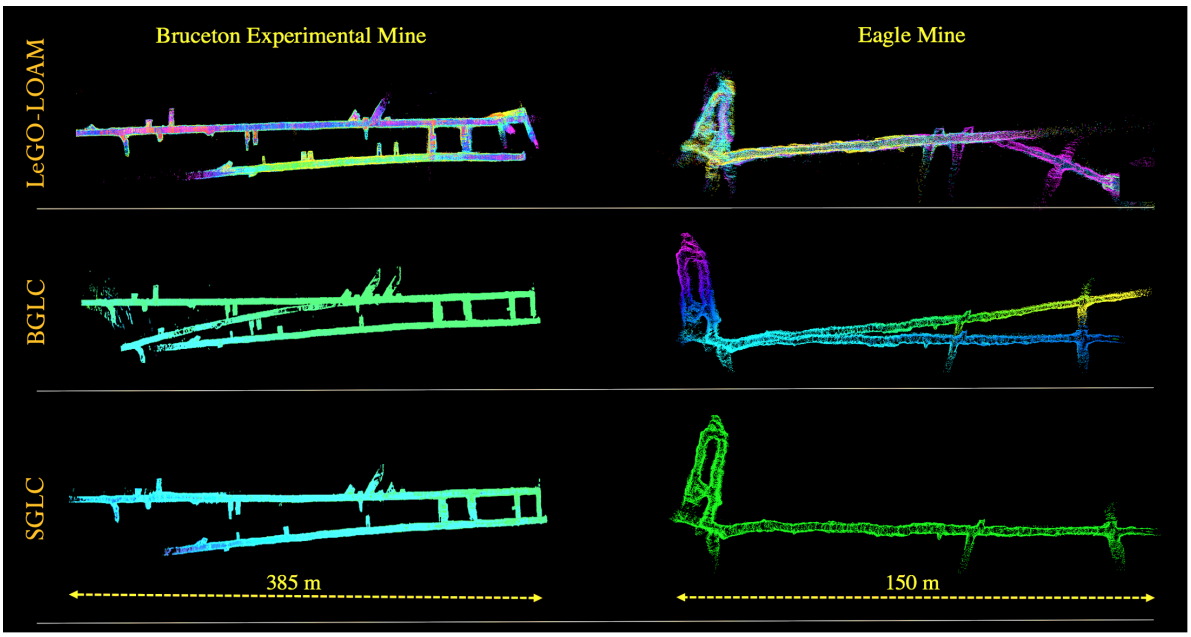

Fig. 5.23: Maps of the Bruceton Experimental mine and the Eagle mine obtained using the LeGO-LOAM, BGLC and proposed SGLC methods.

candidates and thus it is drift-resilient. Fig. 5.24 reports comparison of mapping results based on the Absolute Pose Error (APE) metric for LeGO-LOAM, BGLC and the proposed SGLC methods using datasets obtained from four perceptually-degraded subterranean environments. The box plots are obtained by running each algorithm 10 times on each dataset and recording the localization accuracy of several known landmarks along the robot path. The results show the SGLC method consistently outperforms LeGO-LOAM and BGLC methods across all environments with different levels of perceptual degradation, and thus the drift in estimated robot trajectories is significantly reduced as reflected in small APE values. The results also show all three methods suffer from relatively larger drift in the Experimental and Safety Research mines. This is due to presence of several long, flat and featureless corridors in these mines that leads to more noisy odometric estimates.

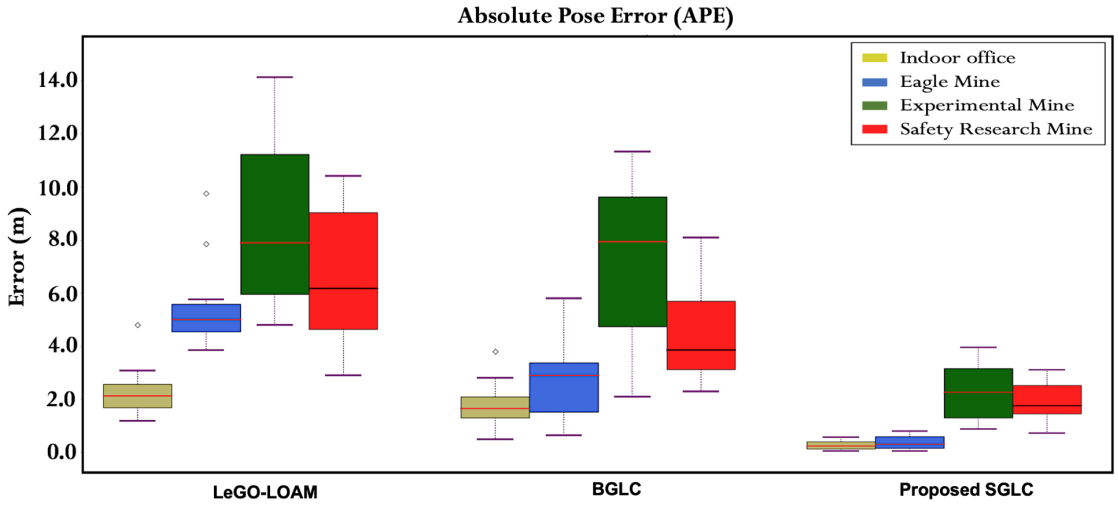

Fig. 5.24: Quantitative comparison of mapping results based on the Absolute Pose Error (APE) using LeGO-LOAM, BGLC and the proposed SGLC methods in 4 perceptually-degraded subterranean environments. The mapping results are obtained by running each method 10 times in each environment

Chapter 6

Conclusion

6.1 Summary

While dramatic progress has been made over the past few decades in the field of single and multi-robot SLAM, localization and mapping in perceptually-degraded environments presents a variety of challenges due to ambiguity of the environment, sparsity of distinctive landmarks and poor illumination. As presented in Chapter 3, in these extreme settings, the front-end is exposed to major challenges including data association ambiguity that can result in accumulation of error and significant drift in the estimated robot trajectory in long-term or large-scale navigation. This highlights the importance of a drift-resilient and robust loop closure detection method that can reduce the drift in the estimated robot trajectory, and enable merging maps obtained by individual robots into a globally consistent map of the environment in a multi-robot SLAM system.

In this thesis, a set of methods and metrics were developed to enable a degeneracy-aware and drift-resilient loop closure detection method, with improved detection rate, robustness and reliability to enable single- and multi-robot autonomous navigation in

unknown and extreme subterranean environments. The degeneracy-aware SLAM front-end developed in Chapter 4 is capable of inferring the level of geometric degeneracy in an unknown environment through eigenanalysis of the solution of the ICP algorithm. This was validated by comparing the results obtained from lidar odometry against the wheel-inertial odometry in cases where the wheel odometry was known to be reliable.

While commonly used geometric loop closure methods periodically attempt loop closures without considering the level of geometric degeneracy of a scene, it is crucial to avoid closing loops in ambiguous areas as it could result in poor estimation of motion between lidar scans which could in turn lead to catastrophic distortions of the map. As shown in real-world experiments, the proposed degeneracy-aware front-end significantly improves the quality and accuracy of obtained 3D maps in both single- and multi-robot SLAM systems by constraining the search for loop closures to areas with high level of observability. Using this method, unobservable areas that could lead to data association ambiguity and spurious loop closures are removed from loop closure consideration. Through analysis of the geometric degeneracy metric, κ, in a variety of indoor and underground settings, it was shown the metric can be reliably used to determine geometric degeneracy in unknown environments.

To overcome the challenge of missed loop closure opportunities commonly encountered in geometric methods, in Chapter 5 a degeneracy-aware and drift-resilient semantic-geometric loop closure detection method was developed to enable pose-invariant determination of loop closures. Using the proposed method the search space for loop closures can be expanded as needed to account for drift, without increasing the probability of spurious loop closures. In the proposed multi-stage method, putative loop closures are identified using a pre-matching step where spatial similarity of the local scenes are evaluated using 2D occupancy grid map matching. Following the detection of loop closure candidates, a geometric verification step is used where the quality of each loop closure

candidate is further evaluated. Finally, by relying on consistency checks outlier loop closures are rejected before optimizing the robot trajectory with the new loop closure constraint. Through extensive real-world experiments and comparisons with state-of-the-art methods, it was shown the proposed degeneracy-aware and drift-resilient loop closure detection method can dramatically improve localization and mapping accuracy, particularly in perceptually-degraded and extreme subterranean environments.

6.2 Future Work

The degeneracy-aware and drift-resilient loop closure detection method developed in this thesis is part of the lidar-based SLAM system of the NeBula autonomy architecture. While the architecture has afforded the CoSTAR team from NASA Jet Propulsion Laboratory the first place in the February 2020 Tunnel Circuit of DARPA Subterranean challenge, there are several grand challenges that need to be overcome to enable coordinated and distributed mapping in a fleet of ground and aerial vehicles with limited communication and computation resources. Future work will focus on developing a robust multi-sensor lidar-centric front-end using the geometric degeneracy detection capability. While in this thesis determination of geometric degeneracy was only used to constrain the search for loop closures to the most observable areas in an unknown environment, the same capability can be used to enable a robust multi-sensor odometry where the front-end evaluates the reliability of lidar odometry in real-time and switches to alternative odometry sources in a geometrically degenerate scene. Another direction for future research is extending this work to a distributed multi-robot SLAM architecture. While the centralized multi-robot architecture presented in the experiments works well for a small team of robots where a base station is in charge of finding the correspondences between robot trajectories and map merging, in large-scale underground

environments with sporadic wireless communication link to a base station, a distributed architecture is required so the robots can exchange information among themselves, and maintain a consistent global map onboard as they explore an unknown environment.

Bibliography

[1] Ellon Mendes, P. Koch, and S. Lacroix. "ICP-based pose-graph slam". *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 195–200, 2016.

[2] María T. Lázaro, R. Capobianco, and G. Grisetti. "Efficient long-term mapping in dynamic environments". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–160, 2018.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. "ORB-SLAM: a versatile and accurate monocular slam system". *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[4] Kamak Ebadi, Y. Change, M. Palieri, A. Stephens, A. H. Hatteland, E. Heiden, A. Thakur, B. Morrell, S. Wood, L. Carlone, and Ali akbar Agha-mohammadi. "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments". *IEEE International Conference on Robotics and Automation*, 2020.

[5] D. Tardioli, L. Riazuelo, D. Sicignano, C. Rizzo, F. Lera, J. L. Villarroel, and L. Montano. "Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments". *Journal of Field Robotics*, 36(6):1074–1101, 2019.

[6] T. Shan and B. Englot. "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain". In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages pp. 4758–4765. IEEE, 2018.

[7] M. Bosse and R. Zlot. "Continuous 3d scan-matching with a spinning 2d laser". In *International Conference on Robotics and Automation (ICRA)*, pages pp. 4312–4319, 2009.

[8] Dorit Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. "Globally consistent 3d mapping with scan matching". *Robotics and Autonomous Systems*, 56(2):130–142, 2008.

[9] Bing-Jui Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and Michael Kaess. "Virtual occupancy grid map for submap-based pose graph slam and planning in 3d environments". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 56(2):2175–2182, 2018.

[10] [Online]. Available: https://velodynelidar.com/vlp 16-lite.html. "Velodyne puck lite lidar".

[11] J. Wang and E. Olson. "Apriltag 2: Efficient and robust fiducial detection". *IEEE International Conference on Intelligent Robots and Systems*, 2016.

[12] Xian-Feng Han, Jesse S. Jin, Ming-Jie Wang, Wei Jiang, Lei Gao, and Liping Xiao. "Loam: Lidar odometry and mapping in real-time". *Signal Processing: Image Communication*, 57:103–112, 2017.

[13] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. "Robust visual inertial odometry using a direct ekf-based approach". *International Conference on Intelligent Robots and Systems*, pages 298–304, 2015.

[14] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. "Keyframe-based visual–inertial odometry using nonlinear optimization". *International Journal of Robotics Research*, 34(3):314–334, 2015.

[15] S. Khattak, F. Mascarich, T. Dang, C. Papachristos, and K. Alexis. "Robust thermal-inertial localization for aerial robots: A case for direct methods". *IEEE International Conference on Unmanned Aircraft Systems*, pages 1061–1068, 2019.

[16] Stephanie Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford. "Visual place recognition: A survey". *IEEE Transactions on Robotics*, 32(1):1–19, 2015.

[17] D. G. Lopez and J. D. Tardos. "Real-time loop detection with bags of binary words". *IEEE International Conference on Intelligent Robots and Systems*, pages 51–58, 2011.

[18] Kamak Ebadi and Ali akbar Agha-Mohammadi. "Rover localization in mars helicopter aerial maps: Experimental results in a mars-analogue environment". *Springer - International Symposium on Experimental Robotics*, pages 72–84, 2018.

[19] Kamak Ebadi and S. Wood. "Scene matching-based localization of unmanned aerial vehicles in unstructured environments". *52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1519–1523, 2018.

[20] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, and W. Churchill. "Navigating, recognizing and describing urban spaces with vision and lasers". *The International Journal of Robotics Research*, 28(11-12):1406–1433, 2009.

[21] H. Qin, M.0 Huang, J. Cao, and X. Zhang. "Loop closure detection in slam by combining visual cnn features and submaps.". *IEEE 4th International Conference on Control, Automation and Robotics*, pages 426–430, 2018.

[22] A. Nuchter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. "6D slam with an application in autonomous mine mapping". *IEEE International Conference on Robotics and Automation*, 2:1998–2003, 2004.

[23] J. Zhang and S. Singh. "Low-drift and real-time lidar odometry and mapping". *Autonomous Robots*, 41(2):401–416, 2017.

[24] I. J. Cox. Blanche. "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle.". *IEEE Transactions on robotics and automationh*, 7(2):193–204, year=1991,.

[25] J. S. Gutmann and C. Schlegel. Amos. "Comparison of scan matching approaches for self- localization in indoor environments". *IEEE Proceedings of the First Euromicro Workshop on Advanced Mobile Robot*, page 61–67, 1996.

[26] F. Lu. "Shape registration using optimization for mobile robot navigation". *PhD thesis - University of Toronto*, 1995.

[27] Yang Chen and Gérard G. Medioni. "Object modeling by registration of multiple range images". *Image Vision Computing*, 10(3):145–155, 1992.

[28] C. Harris and M.J. Stephens. "A combined corner and edge detector". *Alvey Vision Conference*, pages 147–152, 1988.

[29] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. "Fast point feature histograms (FPFH) for 3d registration". *IEEE International Conference on Robotics and Automation*, page 3212–3217, 2009.

[30] Y. Li and E. Olson. "Extracting general-purpose features from lidar data". *IEEE International Conference on Robotics and Automation*, page 1388– 1393, 2010.

[31] G. D. Tipaldi, L. Spinello, and W. Burgard. "Geometrical flirt phrases for large scale place recognition in 2d range data". *IEEE International Conference on Robotics and Automation*, 56(2):2693–2698, 2013.

[32] X. Ji, L. Zuo, C. Zhang, and Y. Liu. "LLOAM: Lidar odometry and mapping with loop-closure detection based correction". *IEEE International Conference on Mechatronics and Automation*, pages 2475–2480, 2019.

[33] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena. "Segmatch: Segment based place recognition in 3d point clouds". *IEEE International Conference on Robotics and Automation*, pages 5266–5272, 2017.

[34] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. "SegMap: 3d segment mapping using data-driven descriptors". *arXiv preprint arXiv:1804.09557*, 2018.

[35] W. Hess, D. Kohler, H. Rapp, and D. Andor. "Real-time loop closure in 2d lidar slam". *IEEE International Conference on Robotics and Automation*, pages 1271–1278, 2016.

[36] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent. "Sparse pose adjustment for 2d mapping". *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[37] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping (slam): Part i". *IEEE robotics and automation magazine*, 13(2):99–110, 2006.

[38] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping (slam): Part ii". *IEEE robotics and automation magazine*, 13(3):108–117, 2006.

[39] R. Smith, M. Self, and P. Cheeseman. "A stochastic map for uncertain spatial relationships". *International Symposium on Robotic Research*, 13(3):467–474, 1988.

[40] R. Smith, M. Self, and P. Cheeseman. "Estimating uncertain spatial relationships in robotics". *Autonomous Robot Vehicles*, 13(3):167–193, 1990.

[41] Brad Schumitsch, S. Thrun, G. Bradski, and K. Olukotun. "The information-form data association filter". *In Advances in Neural Information Processing Systems*, 13(3):1193–1200, 2006.

[42] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyten. "Simultaneous localization and mapping with sparse extended information filters". *International Journal of Robotics Research (IJRR)*, 23(7-8):693–716, 2004.

[43] G. Grisetti, C. Stachniss, , and W. Burgard. "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling". *IEEE International Conference on Robotics and Automation*, page 2432–2437, 2005.

[44] C. Stachniss, G. Grisetti, and W. Burgard. "Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loop". *IEEE International Conference on Robotics and Automation*, page 667–672, 2005.

[45] S. Thrun and M. Montemerlo. "The graph slam algorithm with applications to large-scale mapping of urban structures". *The International Journal of Robotics Research*, 25(5):403–429, 2006.

[46] J. Folkesson and H. I. Christensen. "Graphical slam - a self-correcting map". *IEEE International Conference on Robotics and Automation*, page 383–390, 2004.

[47] F. Lu and E. Milios. "Globally consistent range scan alignment for environment mapping". *Autonomous Robots*, page 333–349, 1997.

[48] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, Z. Liu, H. I. Christensen, and F. Dellaert. "Multi robot object-based slam". *International Symposium on Experimental Robotics*, pages 729–741, 2016.

[49] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert. "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models". *The International Journal of Robotics Research*, 36(12):1286–1311, 2017.

[50] G. Erinc and S. Carpin. "Anytime merging of appearance-based maps". *Autonomous Robots*, 36(3):241–256, 2014.

[51] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. "A tutorial on graph-based slam". *IEEE Intell. Transport. Syst. Mag.*, pages 31–43, 2010.

[52] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "G2O: A general framework for graph optimization". *IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.

[53] M. T. Lazaro, M. P. Lina, P. Pinies, J. A. Castellanos, and G. Grisetti. "Multi robot slam using condensed measurements". *IEEE International Conference on Intelligent Robots and Systems*, pages 1069–1076, 2013.

[54] H. Huang and K. R. Beevers. "Topological map merging". *International Journal of Robotics Research*, 24(8):601–613, 2005.

[55] M. Trentini M. Seto S. Saeedi, L. Paull and H. Li. "Group mapping: A topological approach to map merging for multiple robots". *IEEE Robot. Autom. Mag.*, 21(2):60–72, 2014.

[56] T. M. Bonanni, D. C. Bartolomeo, , and G. Grisetti. "3-D map merging on pose graphs". *IEEE Robotics and Automation Letters*, 2(2):1031–1038, 2017.

[57] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age". *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[58] Pomerleau François, Francis Colas, Roland Siegwart, and Stéphane Magnenat. "Comparing icp variants on real-world data sets". *Autonomous Robots*, 34(3):133–148, 2013.

[59] A. Segal, D. Haehnel, and S. Thrun. "Generalized-ICP". *Robotics: science and systems*, 2(4):435, 2009.

[60] K. Somani Arun, Thomas S. Huang, and Steven D. Blostein. "Least-squares fitting of two 3-d point sets". *IEEE Transactions on pattern analysis and machine intelligence*, 5:698–700, 1987.

[61] Dirk Holz, Alexandru E. Ichim, Federico Tombari, Radu B. Rusu, and Sven Behnke. "Registration with the point cloud library: A modular framework for aligning in 3-d". *IEEE Robotics Automation Magazine*, 22(4):110–124, 2015.

[62] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. "A system for volumetric robotic mapping of abandoned mines". *IEEE International Conference on Intelligent Robots and Systems*, 3:4270–4275, 2003.

[63] D. Tardioli, D. Sicignano, L. Riazuelo, J. L. Villarroel, and L. Montano. "Robot teams for exploration in underground environments". *International Workshop ROBOT 11: Robotica Experimental*, pages 205–212, 2012.

[64] D. Tardioli, L. Riazuelo, T. Seco, J. Espelosín, J. Lalana, J. L. Villarroel, and L. Montano. "A robotized dumper for debris removal in tunnels under construction". *Springer, Cham*, 2017.

[65] R. Zlot and M. Bosse. "Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine". *In Field and service robotics, Springer*, pages 479–493, 2014.

[66] M. Leingartner, J. Maurer, A. Ferrein, and G. Steinbauer. "Evaluation of sensors and mapping approaches for disasters in tunnels". *Journal of field robotics*, 33(8):1037–1057, 2016.

[67] A. Jacobson, F. Zeng, D. Smith, N. Boswell, T. Peynot, and M. Milford. "Semi-supervised slam: Leveraging low-cost sensors on underground autonomous vehicles for position tracking". *IEEE International Conference on Intelligent Robots and Systems*, pages 3970–3977, 2018.

[68] S. Baker and I. Matthews. "Lucas–Kanade 20 years on a unifying framework". *International Journal of Computer Vision*, 56(3):221–255, 2004.

[69] P. J. Besl and N. D. McKay. "A method for registration of 3- d shapes". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[70] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. "New algorithms for 2d and 3d point matching: Pose estimation and correspondence". *Pattern Recognition*, 31(8):1019–1031, 1998.

[71] Y. Tsin and T. Kanade. "A correlation-based approach to robust point set registration". *Springer*, 2004.

[72] C. Stefano, A. Birk, and V. Jucikas. "On map merging". *Robotics and autonomous systems*, 53(1):1–14, 2005.

[73] J. L. Blanco, Javier Gonzalez, and J. A. Fernandez-Madrigal. "A new method for robust and efficient occupancy grid-map matching". *Springer Iberian Conference on Pattern Recognition and Image Analysis*, 2007.

[74] Luca Carlone, Miguel Kaouk Ng, Jingjing Du, Basilio Bona, , and Marina Indri. "Simultaneous localization and mapping using rao-blackwellized particle filters in multi robot systems". *Springer Iberian Conference on Pattern Recognition and Image Analysis*, 63(2):283–307, 2011.

[75] J. L. Blanco, J. Gonzalez, and J. A. Fernandez-Madrigal. "A robust, multi-hypothesis approach to matching occupancy grid maps". *Robotica*, 31(5):687–701, 2013.

[76] J. Zhu, L. Ma S. Du, Y. Zejian, and Z. Qiang. "Merging grid maps via point set registration". *International Journal of Robot Autonomy*, 28(2):180–191, 2013.

[77] J. Zhu, L. Ma S. Du, Y. Zejian, and Z. Qiang. "OctoMap: An efficient probabilistic 3d mapping framework based on octrees". *Autonomous robots*, 34(3):189–206, 2013.

[78] S. Saeedi, T. Michael Trentini, S. Mae, and L. Howard. "Multiple-robot simultaneous localization and mapping: A review". *Journal of Field Robotics*, 33(1):3–46, 2016.

[79] M. A. Fischler and R. C. Bolles. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography". *Communications of the ACM*, page 381–395, 1981.

[80] D. Kakuma, S. Tsuichihara, G. A. G. Ricardez, J. Takamatsu, and T. Ogasawara. "Alignment of occupancy grid and floor maps using graph matching". *IEEE 11th international conference on semantic computing*, pages 57–60, 2017.

[81] Andrea Censi. "An accurate closed-form estimate of icp's covariance". *IEEE international conference on robotics and automation*, pages 3167–3172, 2007.

[82] Ji Zhang, Michael Kaess, and Sanjiv Singh. "On degeneracy of optimization-based state estimation problems". *IEEE International Conference on Robotics and Automation*, 2016.

[83] Sai M. Prakhya, L. Bingbing, Y. Rui, and W. Lin. "A closed-form estimate of 3d icp covariance". *14th IAPR International Conference on Machine Vision Applications*, pages 526–529, 2015.

[84] Martin Brossard, S. Bonnabel, and A. Barrau. "A new approach to 3D ICP covariance estimation". *IEEE Robotics and Automation Letters*, 5(2):744–751, 2020.

[85] Silvère Bonnabel, M. Barczyk, and François Goulette. "On the covariance of icp-based scan-matching techniques". *IEEE American Control Conference*, pages 5498–5503, 2016.

[86] David Landry and F. Pomerleau. "CELLO-3D: Estimating the covariance of ICP in the real world". *IEEE International Conference on Robotics and Automation*, pages 8190–8196, 2019.

[87] Zheng Rong and Nathan Michael. "Detection and prediction of near-term state estimation degradation via online nonlinear observability analysis". *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 28–33, 2016.

[88] Simona Nobili, G. Tinchev, and Maurice Fallon. "Predicting alignment risk to prevent localization failure". *IEEE International Conference on Robotics and Automation*, 5(2):1003–1010, 2018.

[89] Zhang Ji and Sanjiv Singh. "A review of algorithms for filtering the 3d point cloud". *Robotics: Science and Systems*, 2(9), 2014.

[90] Z. J. Yew and G. H. Lee. "DFeat-Net: Weakly supervised local 3d features for point cloud registration". *In European Conference on Computer Vision, Springer, Cham*, pages 630–646, 2018.

[91] X. F. Han, J. S. Jin, M. J. Wang, W. Jiang, L. Gao, , and L. Xiao. "A review of algorithms for filtering the 3d point cloud". *Signal Processing: Image Communication*, (57):103–112, 2017.

[92] V. Ila, J. M. Porta, and J. Andrade-Cetto. "Information-based compact pose SLAM". *IEEE Transactions on Robotics*, 26(1):78–93, 2010.

[93] H. Johannsson, M. Kaess, M. Fallon, and J. J. Leonard. "Temporally scalable visual slam using a reduced pose graph". *IEEE Int. Conf. Robot. Autom.*, pages 54–61, 2013.

[94] H. Kretzschmar, C. Stachniss, and G. Grisetti. "Efficient information theoretic graph pruning for graph-based slam with laser range finders". *International Conference on Intelligent Robots and Systems*, pages 865–871, 2011.

[95] [Online]. Available: https://github.com/MichaelGrupp/evo. "EVO: Python package for the evaluation of odometry and slam.".

[96] R. Szeliski. "Computer vision: algorithms and applications". *Springer Science Business Media*, 2010.

[97] [Online]. Available: https://www.mathworks.com/help/vision/ref/pcregrigid.html. "MATLAB 3d point cloud registration".

[98] J. Zhang, M. Kaess, and S. Singh. "On degeneracy of optimization-based state estimation problems". *IEEE International Conference on Robotics and Automation*, pages 809–816, 2016.

[99] Franco Pavese and Alistair B. Forbes. *"Data modeling for metrology and testing in measurement science"*. 2008.

[100] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. "Geometrically stable sampling for the icp algorithm". *Fourth IEEE International Conference on 3D Digital Imaging and Modeling*, pages 260–267, 2003.

[101] A. Elfes. "Using occupancy grids for mobile robot perception and navigation". *Computer*, 22(6):46–57, 1989.

[102] [Online]. Available: https://www.wiki.ros.org/costmap_2d/hydro/staticmap. "Occuapcny grid maps". *Computer*, 22(6):46–57, 1989.

[103] E. Rublee, V. Rabaud, K. Konolige, , and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". *International conference on computer vision*, pages 2564–2571, 2011.

[104] S. Chanop and R. Hartley. "Optimised kd-trees for fast image descriptor matching". *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[105] Birute Ruzgiene and Wolfgang Förstner. "Ransac for outlier detection". *Geodezija ir kartografija*, 31(3):83–87, 2005.

[106] T. Jost and Heinz Hügli. "Fast ICP algorithms for shape registration". *In Joint Pattern Recognition Symposium, Springer, Berlin, Heidelberg*, pages 91–99, 2002.

[107] M. Toews, C. Wachinger, R. S. J. Estepar, and W. M. Wells. "A feature-based approach to big data analysis of medical images". *In International Conference*

on Information Processing in Medical Imaging, Springer, Cham, pages 339–350, 2015.

[108] P. Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame. "DOOR-SLAM: Distributed, online, and outlier resilient slam for robotic teams". *IEEE Robotics and Automation Letters*, 5(2):1656–1663, 2020.

[109] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan. "Pairwise consistent measurement set maximization for robust multi robot map merging". *IEEE International Conference on Robotics and Automation*, 5(2):2916–2923, 2018.

[110] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona. "A fast and accurate approximation for planar pose graph optimization". *International Journal of Robotics Research*, 33(7):965–987, 2014.

[111] F. Dellaert. "Factor graphs and gtsam: A hands-on introduction". *Georgia Institute of Technology*.

[112] A. Santamaria-Navarro, D. D. Fan R. Thakker, B. Morrell, and A. Agha-mohammadi. "Towards resilient autonomous navigation of drones". *Inter- national Symposium on Robotics Research, Hanoi, Vietnam*, 2019.

[113] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, L. Carlone, C. Guaragnella, , and A. Agha-mohammadi. "LOCUS - A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time". *IEEE Robotics and Automation Letters*, 2020.

[114] A. a. Agha-mohammadi, S. Chakravorty, and N. Amato. "Firm: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurement". *IEEE Robotics and Automation Letters*, 2020.

[115] A. a. Agha-mohammadi, S. Chakravorty S. Agarwal, and N. Amato. "SLAP: Simultaneous localization and planning for physical mobile robots via enabling dynamic replanning in belief space". *arXiv preprint arXiv:1510.07380*, 2015.

[116] S.-K. Kim, R. Thakker, and A.-A. Agha-Mohammadi. "Bi-directional value learning for risk-aware planning under uncertainty". *IEEE Robotics and Automation Letters*, 4(3):2493–2500, 2019.

[117] K. Otsu, S. Tepsuporn, R. Thakker, T. S. Vaquero, J. A. Edlund, W. Walsh, G. Miles, T. Heywood, M. T. Wolf, and A. a. Agha-mohammadi. "Supervised autonomy for communication-degraded subterranean exploration by a robot team". *IEEE Aerospace Conference*, pages 1–9, 2020.

[118] P. Nilsson, S. Haesaert, R. Thakker, K. Otsu, C.-I. Vasile, A.-A. Agha-Mohammadi, R. M. Murray, and A. D. Ames. "Toward specification-guided active mars exploration for cooperative robot teams". *California Institute of Technology*, pages 1–9, 2018.

[119] A.-A. Agha-Mohammadi, E. Heiden, K. Hausman, and G. Sukhatme. "Confidence-rich grid mapping". *International Journal of Robotics Research*, 38(12-13):1352–1374, 2019.

[120] D. D. Fan, A. a. Agha-mohammadi, and E. A. Theodorou. "Deep learning tubes for tube mpc". *arXiv preprint arXiv:2002.01587*, 2020.

[121] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A. a. Agha-mohammadi, and E. A. Theodorou. "Bayesian learning-based adaptive control for safety critical systems". *IEEE International Conference on Robotics and Automation*, pages 4093–4099, 2020.

[122] A. Bouman, M. Ginting, N. Alatur, M. Palieri, D. Fan, T. Touma, T. Pailevanian, S. Kim, K. Otsu, J. Burdick, and A. Agha-mohammadi. "Autonomous spot:long-range autonomous exploration of extreme environments with legged locomotion". *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[123] J. Redmon and A. Farhadi. "Yolov3: An incremental improvement". *arXiv preprint arXiv:1804.02767*, 2018.

[124] [Online]. Available: https://www.subtchallenge.com. "DARPA subterranean challenge".